



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Interpolation Methods and the Accuracy of Lattice-Boltzmann Mesh Refinement

S. M. Guzik, T. H. Weisgraber, P. Colella, B. J. Alder

October 24, 2012

Journal of Computational Physics

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

# Interpolation methods and the accuracy of lattice-Boltzmann mesh refinement

Stephen M. Guzik<sup>a,\*\*</sup>, Todd H. Weisgraber<sup>b,\*</sup>, Phillip Colella<sup>c</sup>, Berni J. Alder<sup>b</sup>

<sup>a</sup>*Department of Mechanical Engineering, Colorado State University, Fort Collins, CO 80523-1374, United States*

<sup>b</sup>*Lawrence Livermore National Laboratory, Livermore, CA 94550, United States*

<sup>c</sup>*Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States*

---

## Abstract

A lattice-Boltzmann model to solve the equivalent of the Navier-Stokes equations on adaptively refined grids is presented. A method for transferring information across interfaces between different grid resolutions was developed following established techniques for finite-volume representations. This new approach relies on a space-time interpolation and solving constrained least-squares problems to ensure conservation. The effectiveness of this method at maintaining the second order accuracy of lattice-Boltzmann is demonstrated through a series of benchmark simulations and detailed mesh refinement studies. These results exhibit smaller solution errors and improved convergence when compared with similar approaches relying only on spatial interpolation. Examples highlighting the mesh adaptivity of this method are also provided.

*Keywords:* Lattice-Boltzmann, Adaptive Mesh Refinement, Poiseuille Flow, Taylor-Green Vortex

---

## 1. Introduction

Over the past twenty years, the lattice-Boltzmann method (LBM) has emerged as a viable alternative to traditional Navier-Stokes solvers [1]. Though it can be derived as a numerical approximation to the classic Boltzmann equation, the LBM has its origins in lattice-gas automata and recovers hydrodynamic behavior in the limit of small Mach and Knudsen numbers. Among its advantages are an efficient algorithm which is straightforward to parallelize, the ease in which complex boundaries can be incorporated, and the possibility to guarantee numerical stability by the implementation of an H-function. The LBM has been successfully applied to variety of flows, including turbulence [2], porous media [3], and hemodynamics [4].

As with any grid-based method, the cost of a lattice-Boltzmann simulation is dictated by the length scale of the smallest features one wishes to resolve. The standard LBM is

---

\*Corresponding author. Tel.: +1 925 423 6349

\*\*Corresponding author. Tel.: +1 970 491 4682

*Email addresses:* [Stephen.Guzik@colostate.edu](mailto:Stephen.Guzik@colostate.edu) (Stephen M. Guzik), [weisgraber2@llnl.gov](mailto:weisgraber2@llnl.gov) (Todd H. Weisgraber)

formulated on a uniform Cartesian grid in either 2D or 3D which limits the accessible range of scales and the complexity of flows it can address. Several extensions to the standard LBM have been developed to enable computation on more complex meshes, including curvilinear [5], unstructured [6], and locally refined grids [7]. Of these approaches, mesh refinement, with domains that consist of a hierarchy of uniform grids, is the most attractive since it preserves the simplicity of the original algorithm by avoiding some of the more elaborate transformations required for unstructured meshes [6]. For the LBM, there are two mesh refinement approaches, node-based, in which the grid points are located at cell corners, and therefore the coarse and a fraction of the fine grid points are co-located, and volume-based, where grid points are cell-centered and are not co-located at any level of refinement. At coarse-fine grid interfaces, appropriate interpolation and reconstruction algorithms must be constructed to properly manage the flow of information from one grid to another while satisfying conservation laws.

In the node-based approach, first developed by Filippova and Hanel [7], continuity between the fine and coarse grids is achieved by rescaling the non-equilibrium component of the distribution function. Improvements to their method removed a potential singularity [8], whereas Kandhai *et al.* [9] proposed rescaling the lattice velocities to couple the meshes for a finite-difference implementation of the LBM. Others adopted the distribution rescaling to multi-block methods, which lack any underlying coarse cells [10, 11]. Rescaling the distribution function does locally preserve mass and momentum between grids at co-located nodes but does not ensure global conservation, in which the total mass and momentum leaving the coarse grid is equal to the mass and momentum entering the fine grid, and vice-versa [12].

Alternatively, the volume-based methods [13, 14] can be formulated to conserve mass and momentum across grid interfaces. Another advantage of the volume-based methods is the ability to incorporate any type of collision operator, whereas the node-based methods are limited to the single relaxation time BGK collision operator [14]. Both piecewise constant [14] and piecewise linear [13] spatial interpolation schemes have been demonstrated to transfer distribution function information from coarse to fine grids, while fine values streaming to the coarse grid are averaged onto the coarse cells.

This paper introduces an alternative cell-centered mesh refinement approach developed within the Chombo adaptive mesh refinement (AMR) framework [15], a C++ library based on the methodology of Colella and Berger [16] for finite-volume computational fluid dynamics (CFD) methods. We propose a space-time interpolation to populate a single layer of ghost cells on each fine grid. We demonstrate that it is fully conservative in mass and more accurate than previous interpolations. We also highlight some of the special considerations that are unique to lattice-Boltzmann grid refinement. Since the original LBM is not a formulation based on control-volumes, there is no flux defined at the faces of the cells surrounding the lattice sites. The consequence of this is that interpolations to fill fine grid ghost cells from the coarse grid need to be fully conservative as conservation cannot be enforced by simply guaranteeing a consistent flux. Another important difference is that in a finite-volume scheme, the fluxes only need to be formulated on the faces of the cells. In contrast, lattice-Boltzmann methods require interpolation and mechanisms for ensuring conservation at each lattice velocity direction near the interface between coarse and fine grids.

As with all AMR schemes, proper care in the construction of the interpolation between grids will ensure the error introduced by the refinement does not excessively affect the solu-

tion error. While some lattice-Boltzmann grid refinement efforts discuss error in a limited context, to our knowledge, there has not been a systematic convergence study of refinement errors along with an examination of how the order of the LBM is affected. We describe and present a thorough error analysis using a series of benchmark calculations to validate our approach and to provide a comparison to other methods.

We describe the single grid lattice-Boltzmann algorithm in section 2, followed by a detailed discussion of our multi-level AMR algorithms in section 3. Several benchmark simulations, including transient Poiseuille flow, the Taylor-Green vortex array, an acoustic pulse, and laminar vortex shedding behind a cylinder are presented in Section 4. Section 5 contains our summary and conclusions.

## 2. Lattice-Boltzmann Method

### 2.1. Single-level algorithm

The problem domain is discretized using a grid,  $\Gamma \subset \mathbb{Z}^D$ , that is a bounded subset of the integer lattice defined by the points  $(j_0, \dots, j_{D-1}) = \mathbf{j} \in \mathbb{Z}^D$  marking the lattice sites. On a Cartesian grid, the cells around each lattice site take the form

$$V_{\mathbf{j}} = [\mathbf{x}_0 + (\mathbf{j} - \frac{1}{2}\mathbf{u})\Delta x, \mathbf{x}_0 + (\mathbf{j} + \frac{1}{2}\mathbf{u})\Delta x], \quad (1)$$

where  $\mathbf{x}_0 \in \mathbb{R}^D$  is some fixed origin of coordinates,  $\Delta x$  is the mesh spacing, and  $\mathbf{u} \in \mathbb{Z}^D$  is the vector whose components are all equal to one.

The lattice-Boltzmann equation describes the evolution of a fictitious single-particle distribution function,  $f_i(\mathbf{j}, t)$ , which is the mass density of particles at time  $t$  moving with lattice velocity  $\mathbf{e}_i$  at site  $\mathbf{j}$  in  $\Gamma$ ,

$$f_i(\mathbf{j} + \mathbf{e}_i\Delta t, t + \Delta t) = f_i(\mathbf{j}, t) + \mathcal{L}_{ij} (f_j(\mathbf{j}, t) - f_j^{eq}(\mathbf{j}, t)) \equiv \hat{f}_i(\mathbf{j}, t). \quad (2)$$

The right side of this equation describes the collision process in which the distribution function relaxes to a local equilibrium  $f_i^{eq}$ , where  $\mathcal{L}_{ij}$  is the linear collision operator and  $\hat{f}_i(\mathbf{j}, t)$  denotes the post-collision state. The most popular version of LB uses the Bhatnagar-Gross-Krook (BGK) collision operator [17],  $\text{diag}(\mathcal{L}) = -1/\tau_\nu$ , with a single relaxation time,  $\tau_\nu$ . This parameter is related to the kinematic viscosity by  $\nu = c_s^2(\tau_\nu - \frac{1}{2})\Delta t$ , where  $c_s = \Delta x/\sqrt{3}\Delta t$  is the speed of sound, a fixed quantity describing the rate at which information propagates across the lattice. Thus, Equation (2) can be decomposed into a collision step followed by a streaming step, in which the distribution functions are advanced to neighboring lattice sites in their respective directions determined by  $\mathbf{e}_i\Delta t$ . The macroscopic hydrodynamic quantities, density ( $\rho$ ), momentum ( $\rho\mathbf{u}$ ), and momentum flux ( $\Pi_{mn}$ ), are computed from moments of the distribution functions,

$$\rho = \sum_i f_i, \quad \rho\mathbf{u} = \sum_i f_i\mathbf{e}_i, \quad \Pi_{nm} = \sum_i f_i\mathbf{e}_i^m\mathbf{e}_i^n. \quad (3)$$

The sums are evaluated over all velocities in the lattice. The lattice is specified by the designation  $DdQn$ , where  $d$  is the dimension and  $n$  the number of discrete velocities. In the D3Q19 lattice considered in this work, the velocity directions extend across faces  $\|\mathbf{e}_i\|_1 = 1$

( $i \in [1, 6]$ ) and across edges  $\|\mathbf{e}_i\|_1 = 2$  ( $i \in [7, 18]$ ), in addition to the rest velocity  $\|\mathbf{e}_i\|_1 = 0$  ( $i = 0$ ). The equilibrium distribution depends on the macroscopic variables and is given by

$$f_i^{eq} = w_i \rho \left( 1 + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right), \quad (4)$$

where the weights  $w_i \in \{1/3, 1/18, 1/36\}$  are for the rest, face, and edge velocities, respectively.

The LB method is an efficient fluid dynamics solver with a straightforward algorithm that avoids additional mathematical complexities, such as solving the Poisson equation for the pressure. It has little numerical diffusion and can achieve near linear scalability in parallel performance [18]. Using a Chapman-Enskog expansion, the lattice-Boltzmann equation can recover the Navier-Stokes equations for small Mach and Knudsen numbers [19]. Within these limits it is second-order accurate in space and time on a uniform grid.

## 2.2. Additional collision operators

### 2.2.1. Multiple relaxation

Lattice-Boltzmann suffers from instabilities as  $\tau_\nu \rightarrow 1/2$  [20] that can be partially suppressed by a more sophisticated class of collision operators: the multiple relaxation time (MRT) models [21]. In the MRT formulation, the collision operator is diagonalizable,  $\mathcal{L} = \mathbf{M}^{-1} \mathbf{S} \mathbf{M}$ , where the rows of  $\mathbf{M}$  are the eigenvectors  $\mathbf{m}^k$  with corresponding eigenvalues  $\lambda_k$  ( $k \in [0, 18]$  for the D3Q19 lattice) of  $\mathcal{L}$ . The eigenvalues contained in the diagonal matrix  $\mathbf{S}$  are associated with a particular relaxation time ( $\lambda_k = -1/\tau_k$ ) and moment  $n^k$  obtained through a linear transformation of the distribution function  $f_i$  using the eigenvectors as the orthogonal basis,

$$n^k = \sum_{i=0}^{18} m_i^k f_i(\mathbf{j}, t). \quad (5)$$

These 19 moments correspond to the density, momentum, the deviatoric components of the momentum flux tensor (the viscous modes), and higher order ghost or kinetic modes. To be consistent with hydrodynamics and the symmetry of the higher moments, the 19 eigenvalues cannot be independent [22]. To conserve mass and momentum, these four hydrodynamic moments have eigenvalues of zero ( $\lambda_k = 0$ ). In our MRT simulations, we incorporate the five relaxation times optimized for stability by Lallemand & Luo [20]. One of the momentum flux moments has a relaxation time  $\tau_\eta$  related to the bulk viscosity,  $\eta = \frac{2}{3} \rho c_s^2 (\tau_\eta - \frac{1}{2}) \Delta t$ , whereas the other flux moments relax at time  $\tau_\nu$ , which defines the kinematic viscosity as before. The three remaining optimized eigenvalues relax the ghost modes.

In practice, rather than constructing the matrix  $\mathcal{L}$ , collisions are computed in the moment space defined by Equation 5. The post-collision states  $\hat{n}^k$  are then transformed back according to

$$\hat{f}_i = w_i \sum_{k=0}^{18} \alpha^k m_i^k \hat{n}^k \quad (6)$$

using unnormalized eigenvectors with an orthogonality condition  $\alpha^k \sum_i w_i m_i^k m_i^\ell = \delta_{k\ell}$ , where  $w_i$  are the weights from the equilibrium distribution [23]. Another version of the MRT collision operator, the two-relaxation time (TRT) method is optimized to reduce the error for wall-bounded flows whose boundaries conform to a sequence of connected cell faces (i.e., a straight or stair-step boundary) [22]. The first eigenvalue is the usual viscous one  $\lambda_1 = -1/\tau_\nu$ , and the second,  $\lambda_2 = \frac{-16\tau_\nu+8}{8\tau_\nu-1}$ , is tuned to keep the solution independent of  $\tau_\nu$ . To avoid stair-stepping, a more contoured boundary can be accommodated by an interpolated bounce-back condition. For this work, we adopted a mass-conserving implementation of the multi-reflection rule [24].

A body force, such as gravity or a pressure gradient, can be incorporated by adding an additional term to the collision operator  $\mathcal{L}_{ij}(f_j(\mathbf{j}, t) - f_j^{eq}(\mathbf{j}, t)) + \mathcal{F}_i$ :

$$\mathcal{F}_i = w_i \left[ \frac{\mathbf{e}_i \cdot \mathbf{F}}{c_s^2} + \frac{\mathbf{M}_F \cdot (\mathbf{e}_i \mathbf{e}_i - c_s^2 \mathcal{I})}{2c_s^4} \right] \Delta t, \quad (7)$$

where  $\mathbf{F}$  is the external force density, and

$$\mathbf{M}_F = \frac{2 + \lambda_\nu}{2} \left[ \mathbf{u} \mathbf{F} + \mathbf{F} \mathbf{u} - \frac{2}{3} (\mathbf{u} \cdot \mathbf{F}) \mathcal{I} \right] + \frac{2 + \lambda_\eta}{3} (\mathbf{u} \cdot \mathbf{F}) \mathcal{I}. \quad (8)$$

For the MRT models, contributions from the external force are added to the momentum and momentum flux moments during the collision process and the momentum is redefined as  $\rho \mathbf{u} = \sum_i f_i \mathbf{e}_i + \frac{1}{2} \mathbf{F} \Delta t$  [25]. For the BGK model, Equation (8) simplifies to

$$\mathbf{M}_F = \frac{2 + \lambda_\nu}{2} (\mathbf{u} \mathbf{F} + \mathbf{F} \mathbf{u}). \quad (9)$$

### 2.2.2. The entropic method

While the MRT collision operator offers improved stability it cannot guarantee it. The entropic lattice-Boltzmann (ELB) method provides unconditional stability by satisfying a discrete analog of the Boltzmann H-theorem:  $\mathcal{H}(t + \Delta t) \leq \mathcal{H}(t)$ , where  $\mathcal{H}$  is the total system entropy function,  $\mathcal{H}(t) = \sum_{j \in \Gamma} H(f)$  and

$$H(f) = \sum_{i=0}^{18} f_i \ln \left( \frac{f_i}{w_i} \right) \quad (10)$$

is the local entropy function [26]. The ELB scheme uses a modified version of the BGK equation

$$f_i(\mathbf{j} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{j}, t) - \frac{\alpha}{2\tau_\nu} (f_j(\mathbf{j}, t) - f_j^{eq}(\mathbf{j}, t)) \quad (11)$$

with a local equilibrium

$$f_i^{eq} = \rho w_i \prod_{d=1}^3 \left( 2 - \sqrt{1 + \left( \frac{u_d}{c_s} \right)^2} \right) \left( \frac{2u_d + \sqrt{1 + \left( \frac{u_d}{c_s} \right)^2}}{1 - u_d} \right)^{e_i^d} \quad (12)$$

which minimizes the entropy function  $H$ . The parameter  $\alpha$  is the positive root of the entropy condition  $H(f + \alpha(f^{eq} - f)) = H(f)$  so then for any  $\tau_\nu > \frac{1}{2}$ , the entropy function will decrease in time, ensuring stability. If the mesh is sufficiently refined,  $\alpha \rightarrow 2$ , and the ELB method recovers the original BGK operator, albeit with a slightly different local equilibrium. Solving for  $\alpha$  and computing the logarithms significantly slows down the collision algorithm by almost an order of magnitude. To help circumvent the additional cost, ELB can be selectively applied to only those cells that exceed a given tolerance  $\delta$  for the deviation from local equilibrium:  $|(f_i^{eq} - f_i)/f_i| > \delta$  [27]. Cells near equilibrium relax with the standard BGK method. The effect of the entropic method is to introduce additional dissipation by moving the post-collision state closer to local equilibrium. Although this adversely affects the accuracy, it is more preferable than an unstable solution.

### 3. Adaptive Mesh Refinement

To implement adaptive mesh refinement (AMR), we make use of the Chombo library for parallel AMR [15] and follow the strategies used therein. For a continuous spatial problem domain, denoted by  $\Gamma$ , the discretization of space is extended to support a hierarchy of refined lattices

$$\Gamma^\ell, \quad \ell = 0 \dots \ell_{\max}, \quad \text{where} \quad \Gamma^{\ell+1} = \mathcal{C}_{n_{\text{ref}}^\ell}^{-1}(\Gamma^\ell),$$

each covering the same spatial domain. In the above,  $\mathcal{C}_{n_{\text{ref}}^\ell}$  is a coarsening operator (with  $\mathcal{C}^{-1}$  performing refinement) and  $n_{\text{ref}}^\ell$  is the refinement ratio between levels  $\ell$  and  $\ell + 1$ , typically a power of two. Adaptive mesh refinement calculations are performed on a hierarchy of nested meshes  $\Omega^\ell \subset \Gamma^\ell$ , with  $\Omega^{\ell+1} \subset \mathcal{C}_{n_{\text{ref}}^\ell}^{-1}(\Omega^\ell)$ , and on the coarsest level,  $\Omega^0 = \Gamma^0$  (i.e., the finer grids are nested inside the coarser grids). The grid levels are considered as overlapping rather than embedded and where they overlap, cells on coarser levels are always completely covered by cells on finer levels. In other words, a cell in  $\Omega^{\ell-1}$  is either completely covered by the cells defined by  $\Omega^\ell$  or its intersection has zero volume. Since meshes on different levels may cover the same region in space, the *valid* solution is provided by the finest mesh in these regions. Regions on a given level are considered *valid* if they are not covered by cells on a finer level, whereas *invalid* regions contain cells covered by finer cells:

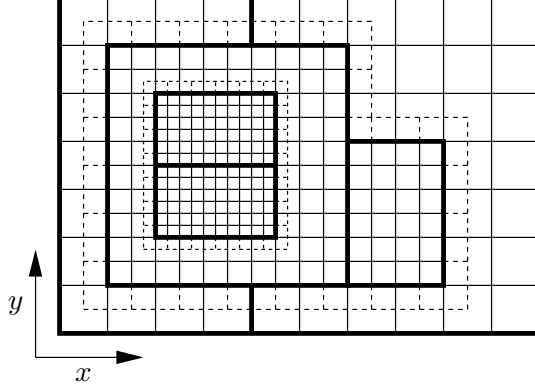
$$\Omega_{\text{valid}}^\ell = \Omega^\ell - \mathcal{C}_{n_{\text{ref}}^\ell}(\Omega^{\ell+1}).$$

Typically,  $\Omega^\ell$  is decomposed into a disjoint union of rectangular boxes,  $\Omega_k^\ell$  where  $\Omega_k^\ell \cap \Omega_{k'}^\ell = \emptyset$  if  $k \neq k'$ , in order to perform calculations in parallel. Each rectangular box,  $\Omega_k^\ell$ , is surrounded by  $q$  layers of ghost lattice sites (or ghost cells),  $\mathcal{G}(\Omega_k^\ell, q)$ . The operator  $\mathcal{G}$  grows a lattice in all directions; defining  $\Omega + \mathbf{j}$  as the translation of a set by a point in  $\mathbb{Z}^D$  and with the max norm  $\|\mathbf{j}\|_\infty = \max(|j_0|, \dots, |j_{D-1}|)$ ,

$$\mathcal{G}(\Omega, q) = \bigcup_{\|\mathbf{j}\|_\infty \leq q} \Omega + \mathbf{j}. \quad (13)$$

The ghost cells permit independent operations on the boxes, support intralevel communication, and allow for the prescription of boundary conditions at the edge of the problem domain. Ghost cells are also characterized as being valid or invalid but with slightly different meaning





**Figure 1** A three level AMR grid with  $n_{\text{ref}} = 2$  and nesting sufficient for two cells to separate level  $\ell + 1$  from level  $\ell - 1$ . A single layer of invalid ghosts cells (only overlapping coarser cells) is shown surrounding the finer two levels. In this drawing, each grid level is decomposed into two boxes which are outlined by thick lines.

than for interior cells. Valid ghost cells overlap actual cells on a level,  $(\mathcal{G}(\Omega_k^\ell, q) - \Omega_k^\ell) \cap \Omega^\ell$ , whereas invalid ghost cells overlies coarser cells or exist outside the problem domain. In general, valid ghost cells are assigned solution values from interior cells on the same level, invalid ghosts within the problem domain are set by interpolating from a coarser level, and invalid ghost sites outside the problem domain are set according to the boundary condition that is to be enforced.

Away from problem-domain boundaries, it is assumed that there is a sufficient number of cells on level  $\ell$  separating the level  $\ell + 1$  cells from the level  $\ell - 1$  cells such that interpolations to fill invalid ghost cells on finer levels can be independently performed. This is more formally given by

$$(\mathcal{G}(\mathcal{C}_{n_{\text{ref}}}^\ell(\Omega^{\ell+1}), p) \cap \Gamma^\ell) \subseteq \Omega^\ell, \quad (14)$$

where  $p$  is the required number of cells for separating two levels. Grid hierarchies that meet the above condition are described as being *properly nested*.

The temporal analogue of proper nesting defines a collection of discrete times,  $t^\ell = t^{\ell-1} + m\Delta t^\ell : 0 \leq m \leq n_{\text{ref}}^\ell$ . Whenever there is refinement in space, there is an equivalent refinement in time. If level  $\ell$  is advanced 1 time step, level  $\ell + 1$  must be advanced  $n_{\text{ref}}^\ell$  time steps to reach the same time value, a process known as subcycling.

In the lattice-Boltzmann algorithm, one layer of ghost cells,  $q = 1$  is required to perform independent streaming operations in a box and allow for the prescription of boundary conditions. In order to perform the interpolation, two cells are required to ensure that the interpolation to invalid ghost cells on level  $\ell + 1$  is independent of level  $\ell - 1$  ( $p = 2$  in equation (14)). The stencils required for the interpolation are given in section 3.5.3. The time-step,  $\Delta t$  and the mesh spacing,  $\Delta x$ , are both reduced by the refinement ratio. Consequently, they are easily known at any grid level. Additionally, the viscous relaxation time in the collision operator must be rescaled to maintain a constant viscosity across all levels of refinement,  $\tau_\nu^{\ell+1} = n_{\text{ref}}^\ell(\tau_\nu^\ell - \frac{1}{2}) + \frac{1}{2}$ . An example three-level grid ( $\ell = 0, 1, 2$ ) with two boxes on each level ( $k = 0, 1$ ) is shown in Figure 1.

### 3.1. Interlevel Operations

Three principle operations are required to perform calculations on a hierarchy of grids:

**Interpolation:** A second-order accurate interpolator is used to transfer information from a coarse level to a finer level. Interpolation is required to fill in data for new fine lattice sites during a regrid and to fill in data for invalid ghost lattice sites surrounding a finer level during the solution update procedure. To ensure a conservative solution, in each velocity direction, the volume-weighted sum of all distributions streaming into the fine grid from the invalid ghost sites must equal the sum of all distributions streaming from the valid coarse sites into the invalid coarse sites underlying the fine grid. In other words, the interpolation itself has to conserve coarse-grid values. This requirement differs from finite-volume CFD approaches where the interpolation need not be conservative because conservation is instead achieved by ensuring that a consistent flux is evaluated on the interface itself. The necessity of a conservative interpolation, coupled with the many directions in which distributions may cross the interface, leads to elaborate interpolation strategies as discussed in section 3.5.1.

**Average:** An averager is used to transfer information from a fine level to a coarse level. The averaging process operates on conserved values and incurs no additional truncation error. Averaging is used to make the solution in invalid coarse lattice sites consistent with overlapping fine sites.

**Stream Correction:** A stream correction is used to ensure that particle distributions streaming across the interface from the fine to coarse grids is consistent between the two levels, and hence, the overall solution is conservative. Whereas distributions streaming *into* the fine grid are made conservative by the interpolator (the sum over fine-grid values equals a coarse-grid value), stream corrections are used to make distributions streaming *out of* the fine grid conservative (the coarse-grid value equals the sum over fine-grid values).

### 3.2. AMR Workflow

A goal espoused in this paper is the full conservation of the particle distributions streaming across the interfaces between fine and coarse lattices. By itself, conservation of the particle distribution implies conservation of mass, but does not say anything about macroscopic properties obtained from higher moments.

The solution over the complete grid hierarchy is advanced by traversing the grids in order of coarse to fine. The typical work-flow for advancing level  $\ell$  is:

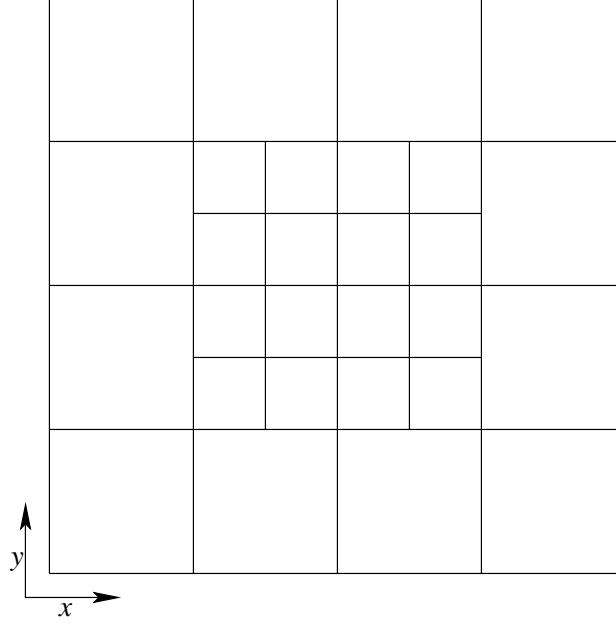
1. Regrid levels finer than  $\ell$  if required. This involves tagging all cells which should compose the finer levels, often based on the magnitudes of solution gradients, and constructing a new, properly-nested mesh hierarchy. In regions where new fine cells appear, the particle distributions are interpolated from the coarser level using a piecewise linear interpolation in each coarse cell.
2. Advance level  $\ell$ .
  - (a) The collision operator is applied to all cells in  $\Omega^\ell$  yielding  $\hat{f}_i(t^\ell)$ .

- (b) Valid ghost sites on level  $\ell$  are filled by copying  $\hat{f}_i(t^\ell)$  from interior sites. Invalid ghost cells within  $\Gamma^\ell$  are assumed to have been filled during operations on level  $\ell - 1$  (see item 3a).
  - (c) Boundary conditions are applied to invalid ghost sites that are outside the problem domain and to any lattice sites within  $\Gamma^\ell$  that are not marked as *fluid cells*. For example, to implement a staircase bounce-back condition in any lattice site representing a wall, one assigns  $\hat{f}_i(\mathbf{j}_{\text{wall}}, t^\ell) = \hat{f}_{i'}(\mathbf{j}_{\text{wall}} + \mathbf{e}_i \Delta t^\ell, t^\ell)$ , where  $\mathbf{e}_{i'} = -\mathbf{e}_i$ .
  - (d) The streaming operator is applied on  $\Omega^\ell$  to obtain  $f_i^\ell(t^\ell + \Delta t^\ell)$ . If a coarser level exists, any particle distributions that streamed from  $\Omega^\ell$  across a coarse-fine interface to a valid cell in  $\Omega^{\ell-1}$  are multiplied by an averaging factor and accumulated to allow for stream corrections. The structures used for accumulating this data are called stream registers and are discussed in section 3.5.4.
3. If a finer level exists, it is sub-cycled or refined in time. In other words,  $n_{\text{ref}}^\ell$  time steps are required to advance level  $\ell + 1$  to time  $t^\ell + \Delta t^\ell$  using  $\Delta t^{\ell+1} = \Delta t^\ell / n_{\text{ref}}^\ell$ . For each sub-cycle  $m = 0 \dots n_{\text{ref}}^\ell - 1$ :
- (a) Interpolate post-collision quantities,  $\hat{f}_i^\ell$ , to the invalid ghost sites surrounding level  $\ell + 1$  at time  $t^\ell + m \Delta t^{\ell+1}$ . These quantities are ready for streaming into the fine mesh. A least-squares algorithm is used to compute the interpolating polynomial in each coarse cell. The interpolation is conservative so that all distributions which stream from level  $\ell$  and cross the coarse-fine interface are fully accounted for. The accounting includes sums of all interpolations to the fine grid over all sub-cycles and leftover values that are accumulated in the stream registers. The details of the conservative space-time interpolation are described in section 3.5.1.
  - (b) Start level  $\ell + 1$  at step 1.
4. If a finer level exists, average the solution down from  $\ell + 1$ . As well, streaming from invalid sites on level  $\ell$  (those underlying level  $\ell + 1$ ) across the coarse-fine interface to valid sites on level  $\ell$  is corrected using the stream registers to match the average of values streaming out from the fine level. The details of the stream corrections are described in section 3.5.4 and are similar to the concept of flux correction for finite-volume methods as described by Berger and Colella [16, 28].
5. The macroscopic fluid properties of density and momentum are determined by taking moments of  $f_i^\ell(t^\ell + \Delta t^\ell)$ .
6. The indices for arrays stored at  $t^\ell$  and  $t^\ell + \Delta t^\ell$  are swapped, discarding the values at the old time and preparing for the next time-step.

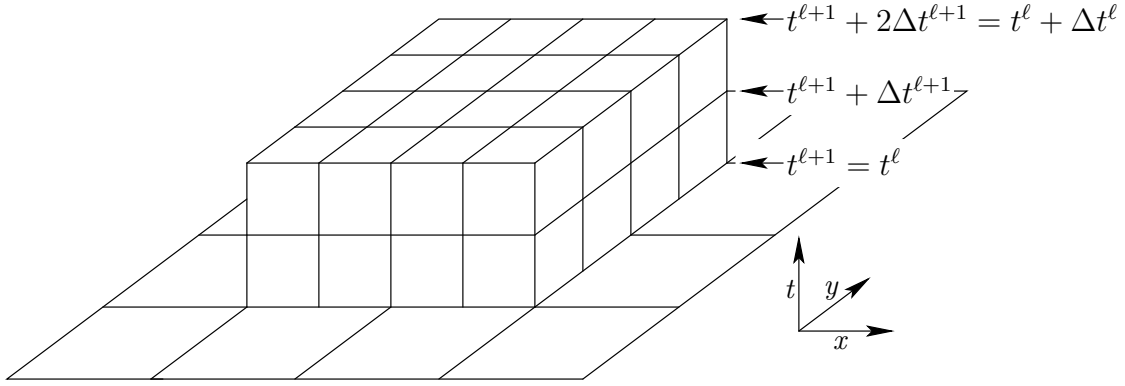
### 3.3. Conservative coupling of grid levels

A representative two-dimensional grid, with a coarse ( $\ell$ ) and fine ( $\ell + 1$ ) level related by a refinement ratio  $n_{\text{ref}}^\ell = 2$ , will be used for the ensuing discussion and is shown in Figure 2. However, it is also important to consider time. In Figure 3, this same grid is shown in space-time with the 2-D coarse cells traced at time  $t^\ell$  and full 3-D space-time fine cells covering two sub-cycles of the algorithm. Note that the lattice sites are at the cell centers of the 2-D spatial grid and on the constant-in-time faces of the 3-D space-time grid.

There are two conservative cell-based approaches for coupling a fine and coarse level, analogous to specifying an initial-value problem or a boundary-value problem. The approach

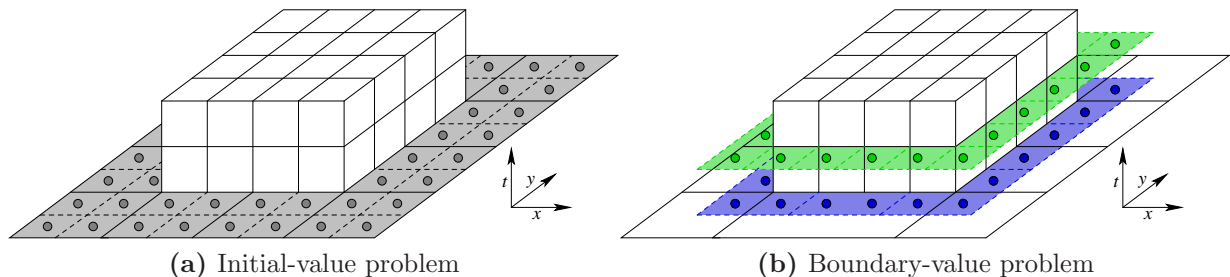


**Figure 2** Two level AMR grid in two spatial dimensions. The refinement ratio between the two levels is  $n_{\text{ref}}^{\ell} = 2$ .



**Figure 3** Two level AMR grid in three dimensions (two spatial and one temporal). The fine cells are traced in space-time over two sub-cycles of the algorithm.

taken by Chen et al. [13] is to consider the interface as an initial-value problem (IVP) as shown in Figure 4a. Sufficient ghost lattice-sites are filled at the base time  $t^{\ell+1} = t^{\ell}$  so that the domain of dependence is fully populated for all subcycles on the fine lattice. For a standard lattice-Boltzmann scheme, with streaming only to neighboring lattice sites, the domain of dependence requires  $n_{\text{ref}}^{\ell}$  layers (or rings) of ghosts cells. An alternative approach, widely used for CFD but also previously used for lattice-Boltzmann methods [10], that was developed as part of this work is to treat the fine lattice as a boundary-value problem (BVP). In this approach, only one layer of ghost cells is required, but it must be filled with new information before each subcycle on the fine lattice. The latter approach has some advantages. In particular, the IVP approach may perturb a steady-state solution as the interpolated distributions are marched across the interface at different times. For a steady



**Figure 4** The solution on the finer-level requires information at the interface to be well-posed. The additional information may take the form of an initial value problem (4a) or a boundary-value problem (4b).

solution, one would expect the same value to appear in ghost sites adjacent to the interface for each subcycle on the fine lattice; however, a solely spatial interpolation will not guarantee this. Chen et al. [13] discuss strategies for avoiding this problem, at the expense of reduced interpolation accuracy. The BVP approach does not suffer from this problem as long as time derivatives are set to zero when the solution is indeed steady. The BVP approach provides the additional advantage of regular box sizes. Every box has one layer of ghost cells, to be filled either by exchange of information from neighboring boxes on the same grid level or by interpolation from a coarser level. Consequently, one can define a fixed box size that is optimized either for memory caches or accelerator architectures (such as graphics processing units) and more easily load-balanced across distributed-memory architectures.

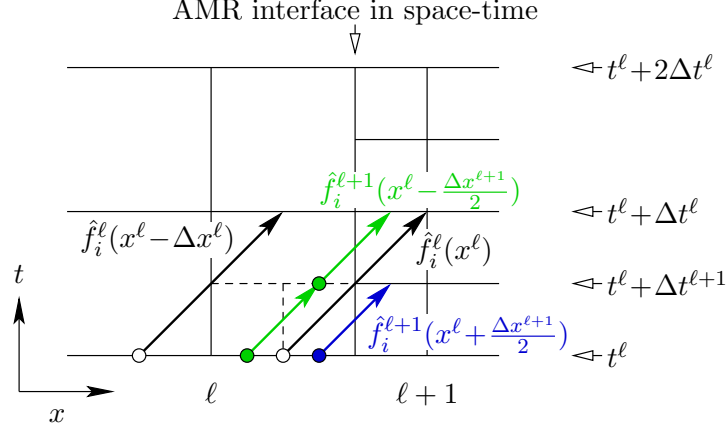
The details of the interpolation and stream corrections are quite intricate in two-spatial dimensions. Therefore, the differences between the IVP and BVP approaches will be first explained for grids with one spatial dimension in section 3.4. A read through section 3.4 is sufficient for understanding the concepts. However, finding a conservative interpolation is trivial with only one spatial dimension. It is necessary to consider two spatial dimensions to illustrate the complexities introduced when particle distributions stream in edge velocity directions,  $\|\mathbf{e}_i\|_1 = 2$ , across grid interfaces (streaming orthogonal to the grid interfaces,  $\|\mathbf{e}_i\|_1 = 1$ , is not much more complex than in one spatial dimension). The interpolation and stream correction strategies for grids with two spatial dimensions are discussed in section 3.5. Section 3.5 provides sufficient information to reproduce the results contained herein. In that section, the challenges of finding conservative interpolation patterns and the reasons for using constrained least-squares approximations to the interpolations become apparent.

### 3.4. Conservation at Refinement Interfaces In One-Spatial Dimension

For both of the IVP and BVP approaches, there is an assumption that the distributions are locally characteristic, e.g.,

$$\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial x} = 0. \quad (15)$$

This implies that the interpolations are performed in one dimension less than the full space-time problem dimension. In the IVP approach, all interpolations are performed in space only. An illustration of the IVP interpolation is shown in Figure 5 for a domain with one spatial dimension and a refinement ratio of two. Space is along the horizontal axis with the

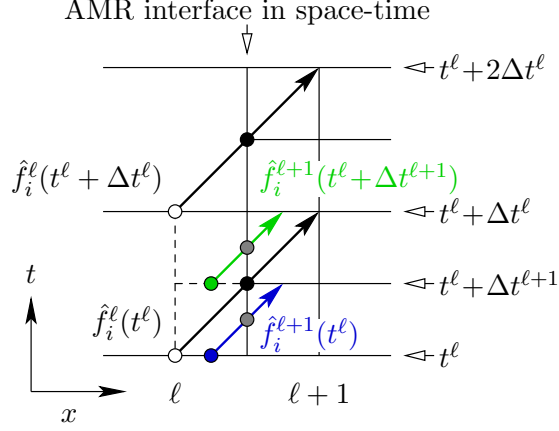


**Figure 5** IVP approach for interpolating across the space-time interface between a coarse and fine grid. The domain has one spatial dimension and the interpolation is performed only in the spatial direction.

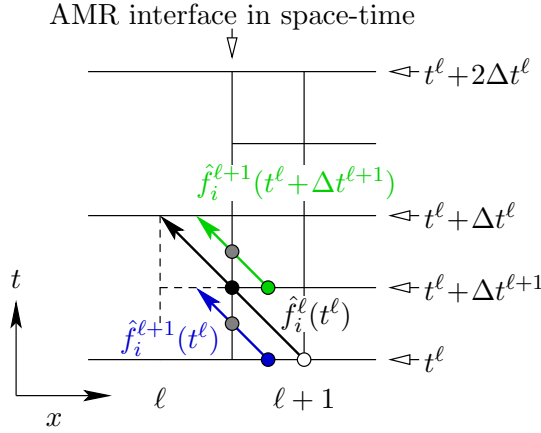
coarse grid on the left and the fine grid on the right. Time is along the vertical axis; two steps on the fine grid are required to reach the same time as one step on the coarse grid. We start knowing  $\hat{f}_i^\ell(t^\ell)$ . The black arrows, hereon labeled stream vectors, show the space-time streaming of a particle distribution, from an origin (white dot) to a destination in velocity direction  $i$ . The two post-collision states, shown by white dots, are used to construct the interpolating polynomial. Before advancing the solution on the fine grids, two layers of ghost cells are filled by interpolating in space at time  $t^\ell$  (green and blue dots).

In the BVP approach developed herein, the interpolation would only be performed in time as shown in Figure 6. The coarse grid is advanced before the fine grid and therefore the solution there is known at times  $t^\ell$  and  $t^\ell + \Delta t^\ell$ , thus enabling interpolation in time. There is only one layer of ghost cells and it must be filled before each advance on the fine level (at times  $t^\ell$  and  $t^\ell + \Delta t^{l+1}$ ). Because the distributions are assumed to be characteristic, the stream vectors can be related to each other in different ways. Our postulate is that the stream vectors should be related to each other at the point in which they cross the coarse-fine AMR interface rather than at their origins. In Figure 6, the distances between the coarse and fine stream vectors are the distances between the black and gray dots; in one-spatial dimension, there is only a difference in time. Note that interpolation with a linear function ( $\frac{\partial \hat{f}_i}{\partial t}$  is known from the coarse solution at the two black dots) is conservative since the gray dots are symmetrically distributed about the black dot.

Both approaches require correction of the distributions on the coarse grid (those that stream from invalid coarse cells underneath the fine grid to valid coarse cells) by the distributions streaming out of the fine grid. For the IVP approach, these distributions propagate into the ghost cells. After the fine grid has been advanced to the same time as the coarse grid, they can simply be averaged to replace the coarse grid solution underneath. In the BVP approach, it is instead necessary to accumulate the distributions leaving the fine grid after each advance on the fine grid. As shown in Figure 7, this is basically the opposite of interpolating. The fine-grid distributions (green and blue arrows) accumulate in a *stream register* and are eventually used to replace the coarse-grid distribution (black arrow).



**Figure 6** BVP approach for interpolating across the space-time interface between a coarse and fine grid. The domain has one spatial dimension and the interpolation is performed only in the temporal direction. The interpolation is performed on the AMR interface between the gray and black dots.



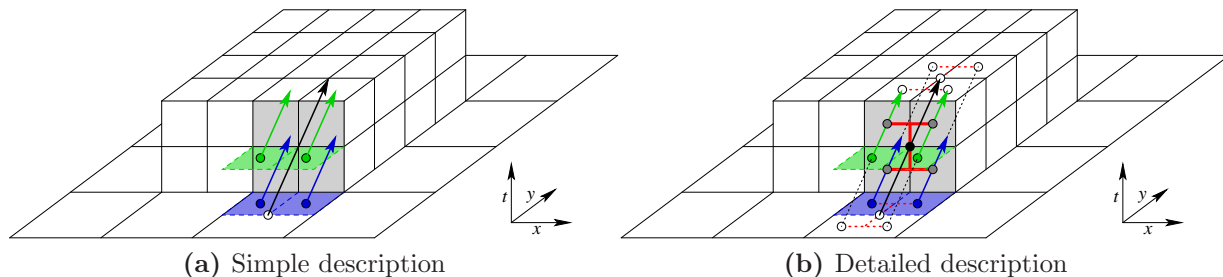
**Figure 7** Corrections to streaming across the space-time interface between a fine and coarse grid. A sum of the distributions streaming from the fine mesh (green and blue) replaces the distribution from an invalid cell on the coarser level (black).

Some differences between the two approaches were already highlighted in section 3.3. In particular, in Figure 5, it is quite obvious how gradients in the  $x$ -direction (differences between the solution in the green and blue dots) translate into gradients in the  $t$ -direction, thus perturbing a steady-state solution. However, perhaps the most important difference between the IVP and BVP strategies is that interpolated distributions in the IVP approach are not relaxed as they propagate through the ghost cells during subcycling. This is shown by the green dot at time  $t^l + \Delta t^{l+1}$  in Figure 5. In the BVP approach, on the other hand, the collision operator can influence the interpolation through gradients in time.

### 3.5. Conservation at Refinement Interfaces In Two-Spatial Dimensions

In the present work, only the D3Q19 lattice is considered. Because velocity directions extend only across faces and edges, the interpolation strategies can instead be fully explained





**Figure 8** Interpolation from coarse to fine for streaming in the face velocity direction  $(0, +y)$ . In Figure 8b, the origin and terminus of all streaming vectors are traced to times  $t^\ell$  and  $t^\ell + \Delta t^\ell$ . The location where each vector crosses the space-time interface between the coarse and fine grid are shown and connected by the red line to describe the interpolation pattern.

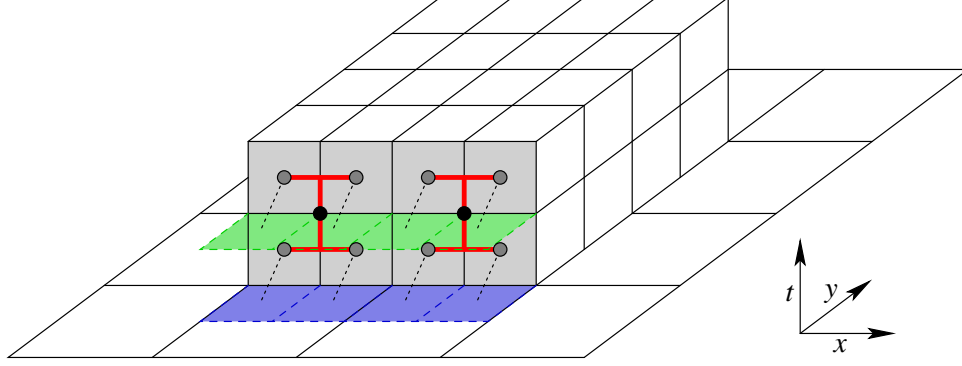
in two dimensions on a D2Q9 lattice where velocity directions extend across the four edges and four corners of a quadrilateral. The space-time grid shown in Figure 3 is used for this purpose.

### 3.5.1. Interpolation Patterns

The interpolation concepts described in Figure 6 are now shown for two spatial dimensions. There are different interpolation patterns depending on whether the streaming is across faces,  $\|\mathbf{e}_i\|_1 = 1$ , or edges,  $\|\mathbf{e}_i\|_1 = 2$ , and whether the coarse cell is adjacent to either faces, an edge forming a convex corner, or an edge forming a concave corner of the fine grid. To assist with deciphering the interpolation patterns, both simple and detailed descriptions are provided. The simple description shows the coarse streaming vector and the ghost sites that are filled by an interpolation centered around the coarse vector, blue for the first subcycle and green for the second subcycle on the fine lattice. Fine streaming vectors from the ghost lattice sites are also shown in blue and green. A portion of the space-time interface between the coarse and fine grids that is relevant to a particular interpolation is shaded in gray. The detailed description adds more information. The fine streaming vectors are traced to locations in the coarse cells containing the origin and terminus of the coarse streaming vector. Gray and black dots respectively illustrate where the fine and coarse streaming vectors cross the interface. Representative patterns that relate the fine and coarse streaming vectors are shown in red on both the interface and the original and terminal coarse cells. These red patterns are collected and interlaced in later figures to show how all ghost sites can be filled for a particular velocity direction. It is worthwhile to note that all fine streaming vectors can be traced to lattices sites (shown by white dots) corresponding to a refinement of the original and terminal cells on the coarse grid.

In Figure 8, a relatively straightforward interpolation pattern is shown for a face velocity  $(0, +y)$  on the  $y_{\min}$  space-time face of the fine grid. In the detailed description of Figure 8b, the locations connected by a solid red line indicate where each streaming vector crosses the interface (forming an “I” pattern). It is immediately obvious that the vectors streaming from the fine ghost cells (blue and green) are symmetrically distributed around the vector streaming from the coarse cell (black) on this space-time boundary (symmetry planes are orthogonal to the  $x$  and  $t$  directions). That the symmetry planes are aligned with the

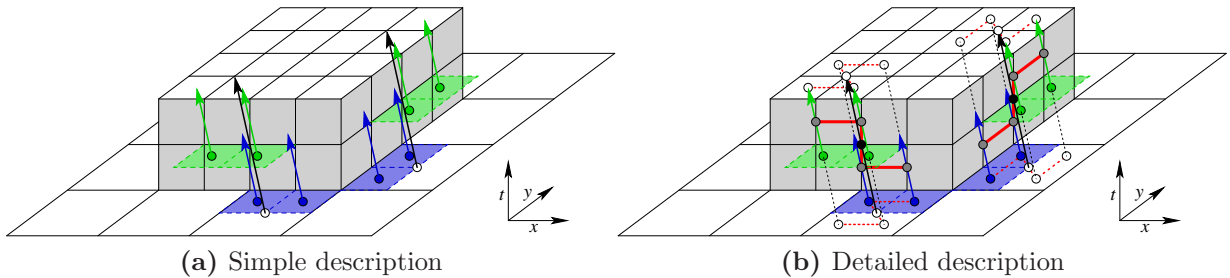




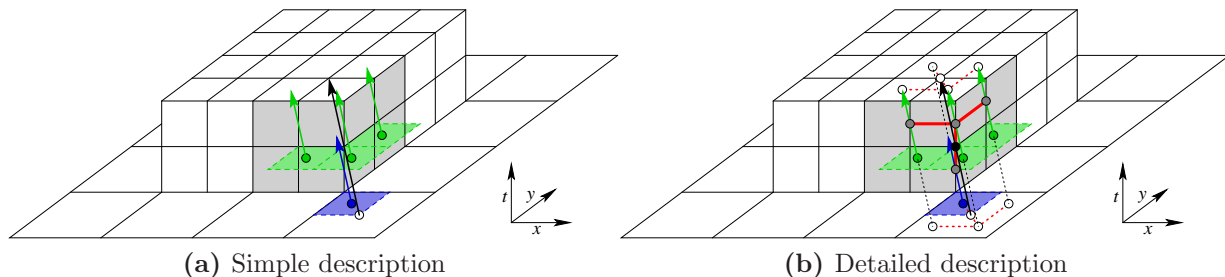
**Figure 9** Interpolation patterns for streaming in the face-velocity direction  $(0, +y)$  for the complete  $y_{\min}$ -boundary of the fine box. The dashed black lines trace backwards from the space-time interface to the ghost lattice sites.

coordinate directions means that a bilinear interpolation will be conservative without any additional effort. Interpolating to all ghost lattice sites in the  $(0, +y)$  direction results in the pattern shown in Figure 9 for the complete  $y_{\min}$  face of the fine box.

Slightly more complicated patterns emerge for interpolating in edge velocity directions on the faces of the fine grid. Figure 10 illustrates the interpolation patterns for the  $(-x, +y)$  velocity direction on both the  $y_{\min}$  and  $x_{\max}$  faces of the fine-grid box. In these cases, connections between the locations where the streaming vectors cross the interface form either “S” or “Z” patterns as shown in red. The gray dots are still symmetrically arranged around the black dot, but symmetry planes are no longer aligned with the coordinate directions. Consequently, a bilinear interpolating polynomial will not automatically satisfy the conservation requirement. To ensure conservation, a constrained least-squares algorithm is used to compute the interpolating polynomial and is discussed in section 3.5.2. However, note that if the time dimension is collapsed, then a symmetry plane perpendicular to the refinement interface is apparent. The implication of this is that a conservative interpolant will not perturb a steady-state solution and is discussed more in section 3.5.2.



**Figure 10** Interpolation from coarse to fine for streaming in the edge velocity direction  $(-x, +y)$  across both the  $y_{\min}$  and  $x_{\max}$  faces of the fine-grid box. In Figure 10b, the origin and terminus of all streaming vectors are traced to times  $t^\ell$  and  $t^\ell + \Delta t^\ell$ . The location where each vector crosses the space-time interface between the coarse and fine grid are shown and connected by the red line to describe the interpolation pattern.



**Figure 11** Interpolation from coarse to fine for streaming in the edge velocity direction  $(-x, +y)$  across the convex  $(x_{\max}, y_{\min})$  corner of the fine-grid box. In Figure 11b, the origin and terminus of all streaming vectors are traced to times  $t^\ell$  and  $t^\ell + \Delta t^\ell$ . The location where each vector crosses the space-time interface between the coarse and fine grid are shown and connected by the red line to describe the interpolation pattern.

For the same edge velocity direction,  $(-x, +y)$ , but at the coarse lattice site just outside the  $(x_{\max}, y_{\min})$  corner of the fine box, a modification to the interpolation pattern is required. Figure 11 illustrates a convex corner of the fine grid. Fine-grid lattice sites that must be filled extend in both the  $-x$  and  $+y$  direction from the corner. The interpolant therefore requires an extra independent variable. The resulting interpolation pattern, shown in red, is in the shape of a “T”. At this location, even if the time-dimension is collapsed, the symmetry is not aligned with a coordinate direction (rather it is orthogonal to the vector  $(x = 1, y = 1)$ ). Conservation is thus not automatically achieved and to counter an imbalance between fine ghost sites in the  $-x$  and  $+y$  directions (along the horizontal part of the “T”), the least-squares algorithm could induce a gradient in the ghosts sites forming the vertical part of the “T”. This would perturb a steady state solution. As discussed in section 3.5.2, the addition of another constraint to the least-squares procedure avoids this problem.

If all ghosts cells required for streaming in the  $(-x, +y)$  velocity direction are filled for the fine box, the interpolation patterns fit together and the arrangement would appear as shown in Figure 12. Interpolations to the yellow locations are not required for streaming into the fine box. However, these values are required for *stream corrections* to ensure conservation as discussed in section 3.5.4.

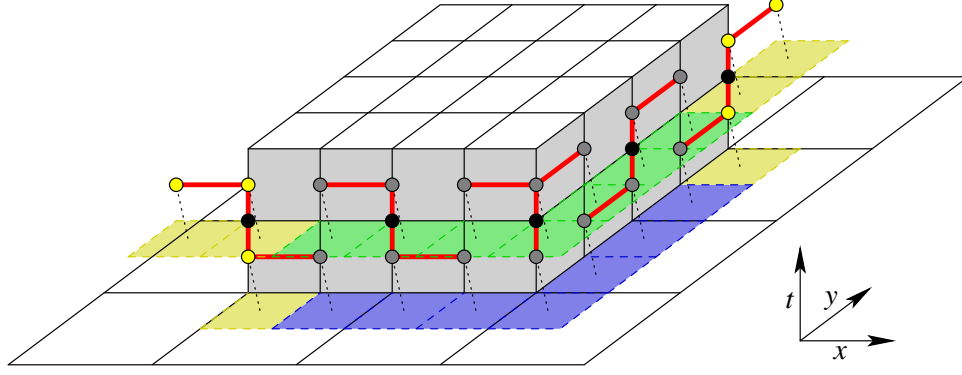
If the corner is concave instead of convex, still for the  $(-x, +y)$  velocity directions, then the interpolation patterns appear as shown in Figure 13. The “T” simply changes to being upside-down.

### 3.5.2. Constrained Least-Squares Interpolation Algorithm

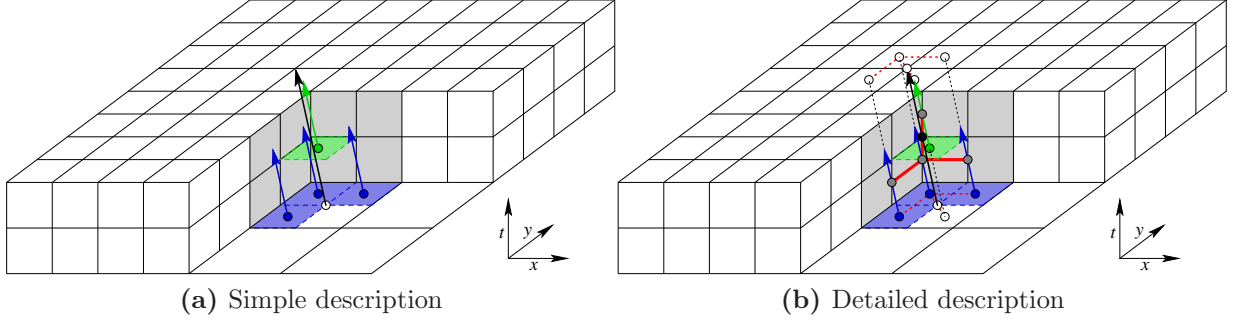
For coarse cells adjacent to the faces of the coarse-fine interface, the interpolation is performed in a canonical space-time coordinate system,  $(t, x, z)$ , where  $z$  is the third spatial dimension not shown in Figs. 2 to 13. Consequently, in the following text, the lattice indices  $\mathbf{j}$  and  $\mathbf{k}$  have an index representing time and two representing space. The interpolating  $n$ -term polynomial in the space-time is given by

$$[1, y, z, t, zt, yt]\boldsymbol{\xi} = \hat{f}_i, \quad (16)$$

consisting of linear components and bilinear components of time and the individual spatial directions. The coefficients (derivatives of the Taylor polynomial) are given by  $\boldsymbol{\xi}$  and are



**Figure 12** Interpolation from coarse to fine for streaming in the edge-velocity direction  $(-x, +y)$  into the fine box. The dashed black lines trace backwards from the space-time interface to the ghost lattice sites. Interpolations to the yellow ghost sites, although provided by a regular application of the interpolation patterns, are not used in the update procedure on the fine grid.



**Figure 13** Interpolation from coarse to fine for streaming in the edge velocity direction  $(-x, +y)$  across the concave  $(x_{\max}, y_{\min})$ , corner of the fine-grid box. In Figure 13b, the origin and terminus of all streaming vectors are traced to times  $t^\ell$  and  $t^\ell + \Delta t^\ell$ . The location where each vector crosses the space-time interface between the coarse and fine grid are shown and connected by the red line to describe the interpolation pattern.

evaluated by solving the system

$$[1, y, z, t, zt, yt]_{\mathbf{k}^\ell} \boldsymbol{\xi}_{j^\ell} = \hat{f}_{i, \mathbf{k}^\ell}, \quad \mathbf{k}^\ell \in \mathcal{I}(j^\ell) \quad (17)$$

using a least-squares algorithm. The stencil around lattice site  $j^\ell$  is denoted by  $\mathcal{I}$ . This system is subject to the constraint

$$\frac{\sum_{j^{\ell+1} \in \mathcal{I}^{-1}(j^\ell)} [1, y, z, t, zt, yt]_{j^{\ell+1}} \boldsymbol{\xi}_{j^{\ell+1}}}{(n_{\text{ref}}^\ell)^{\mathbf{D}}} = \hat{f}_i(j^\ell), \quad (18)$$

where  $\mathcal{I}^{-1}(j^\ell)$  is used to indicate all the destination fine-grid ghost lattice sites in space-time that are filled using a polynomial constructed at  $j^\ell$ . These destinations result from the patterns of interpolation as shown in Figs. 8a, 10a, 11a, and 13a and each one consists of  $(n_{\text{ref}}^\ell)^{\mathbf{D}}$  lattice sites.

In general, for a stencil with  $m$  coarse cells, an equality-constrained least-squares problem can be written as

$$\min_{\mathbf{B}\boldsymbol{\xi}=\mathbf{d}} \|\mathbf{A}\boldsymbol{\xi} - \mathbf{b}\|_2, \quad (19)$$

where  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{p \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ , and  $\mathbf{d} \in \mathbb{R}^p$ . The term  $\mathbf{B}\boldsymbol{\xi} = \mathbf{d}$  describes the single constraint (18) with  $p = 1$ .

For all patterns of interpolation destinations on the faces of the fine box (Figure 8 and 10), if the time dimension is collapsed then planes of symmetry in the patterns align with relevant spatial coordinate directions. This implies that if the solution is steady (not varying in time), the least-squares procedure can easily find a solution that satisfies the conservation constraint *and* guarantees that all coefficients involving a derivative  $\frac{\partial}{\partial t}$  are zero. Consequently, the interpolation will not perturb a steady solution.

Unfortunately, this feature is not retained at the edges of the fine box. There, the patterns of interpolation destinations, when collapsed in time, do not feature symmetry planes that align with coordinate directions (see Figs. 11 and 13). Adopting the procedure described for box faces would produce gradients in time as the least-squares algorithm attempts to satisfy the conservation constraint. Some modifications are necessary.

On coarse cells adjacent to the corners of a fine box, the interpolation is performed in full 4-D space-time,  $(t, x, y, z)$  where again  $z$  is the third spatial dimension not shown in Figs. 2 to 12. The lattice indices  $\mathbf{j}$  and  $\mathbf{k}$  now have an index representing time and three representing space. The interpolating  $n$ -term polynomial in the space-time is given by

$$[1, x, y, z, t, zt]\boldsymbol{\xi} = \hat{f}_i, \quad (20)$$

Here, only one bilinear component of time and the  $z$  direction is retained to avoid a poorly conditioned system. Two constraints are then applied to the least squares system. The first is similar to that shown in (18) and ensures the interpolation is conservative

$$\sum_{\mathbf{j}^{\ell+1} \in \mathcal{I}^{-1}(\mathbf{j}^{\ell})} [1, x, y, z, t, zt]_{\mathbf{j}^{\ell+1}} \boldsymbol{\xi}_{\mathbf{j}^{\ell}} = \hat{f}_i(\mathbf{j}^{\ell}). \quad (21)$$

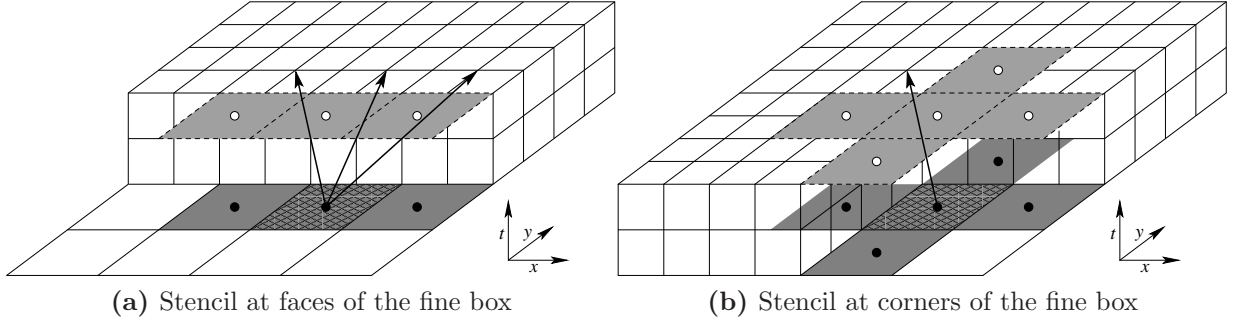
The second insists that the time-derivative in the polynomial is identical to the difference in  $\hat{f}_i(\mathbf{j}^{\ell})$  observed in the coarse time-step,

$$[0, 0, 0, 0, 1, 0]\boldsymbol{\xi}_{\mathbf{j}^{\ell}} = \frac{\partial \hat{f}_i(\mathbf{j}^{\ell})}{\partial t} = \frac{\hat{f}_i(\mathbf{j}^{\ell} + (\Delta t, 0, 0, 0)) - \hat{f}_i(\mathbf{j}^{\ell})}{\Delta t} \quad (22)$$

These modifications ensure that, in the case of a steady flow, all coefficients involving a derivative  $\frac{\partial}{\partial t}$  are zero; conservation is maintained only by balancing gradients of  $x$  and  $y$ . The remainder of the least-squares procedure is the same as discussed for faces of the box except for considerations that  $p$  is now equal to two in Eq. (19).

### 3.5.3. Coarse-Grid Stencils

Stencils that are used for constructing the interpolants are shown in Figure 14. The stencil in Figure 14a is used for all coarse streaming vectors originating in a coarse lattice site that is adjacent to a face of the fine box. As gradients are not required orthogonal



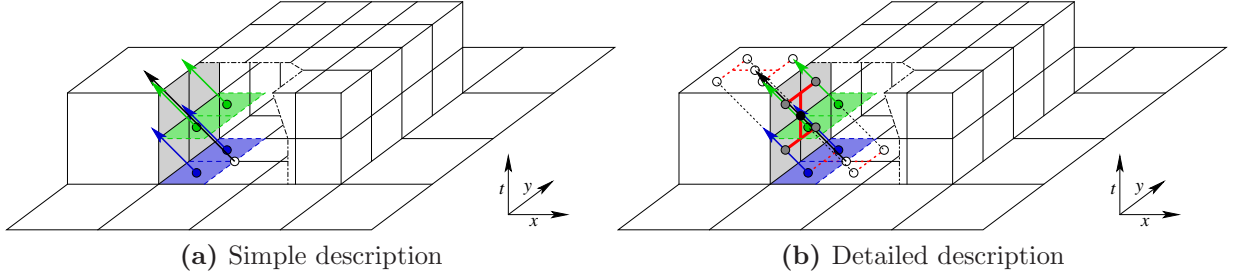
**Figure 14** Stencils for constructing the interpolant on a D2Q9 grid. Coarse lattice sites are marked with solid black dots at time  $t^\ell$  and white dots at time  $t^\ell + \Delta t^\ell$ . The interpolation is centered around a coarse streaming vector which originates at the crisscrossed cell. The stencil in Figure 14a is used for both face and edge velocity directions on the faces of the fine grid. The stencil in Figure 14b is used for edge velocity directions across both convex and concave (shown) corners.

to the face (see space-time discussion regarding Figure 6), there are no stencil cells in that direction. In the 3-dimensional code, there are an additional four stencil cells, two at each time-level neighboring the center of the stencil in the  $+z$  and  $-z$  directions. The 3-D stencil then has 10 coarse cells in total ( $m = 10$  in Eq. (19)). At corners of the fine grid, where only edge-velocity directions are relevant, the interpolant requires gradients in all spatial and temporal directions. The stencil in 2-D forms a plus symbol at each time, centered around the origin of the coarse streaming vector. As for faces, four additional cells are required in the  $z$ -direction for 3-D solutions, making the total number of coarse cells in the stencil,  $m = 14$ .

#### 3.5.4. Stream Registers and Conservation

In section 3.5.1, methods of interpolation were discussed that allow for particle distributions to stream from the coarse to fine lattice in a conservative manner. In this section, the similar but opposite problem of ensuring that the particle distributions streaming from the fine to coarse lattice remain conservative is addressed. In the proposed AMR algorithm (see section 3.2), the coarse grid is everywhere advanced one time step before beginning to advance the finer levels. This provides an approximation (to the resolution of the coarse lattice), of what should stream from invalid coarse cells (those covered by the fine lattice) to coarse lattice sites adjacent to the interface between the coarse and fine grids. However, a more accurate approximation is to instead use a sum of the particle distributions streaming out of the fine lattice. Following terminology and programming strategies used for finite-volume methods, we describe this process of replacing the coarse approximation with summations from the fine lattice as *stream correction* and accumulate the correction in arrays known as *stream registers*. A stream register, providing corrections from level  $\ell + 1$  to  $\ell$ , is denoted by  $\delta f_i^\ell(\mathbf{j}^\ell)$  and exists at any coarse lattice site adjacent to and outside the fine level in velocity direction  $i$ , i.e.,

$$\mathbf{j}^\ell \in \left[ \Omega_{\text{valid}}^\ell \cap \left( \mathcal{C}_{n_{\text{ref}}}^\ell (\Omega^{\ell+1}) + \mathbf{e}_i \right) \right]. \quad (23)$$



**Figure 15** Contributions to the stream register from the fine grid for streaming in face-velocity direction  $(-x, 0)$ . In Figure 15a, lattice sites at the interior of the fine grid that contribute to the stream register associated with the black stream vector are shown in blue for the first subcycle and green in the second subcycle. The trace of fine stream vectors originating at these sites all terminate in a refinement of the cell in which the coarse stream vector terminates; this is how these sites were identified. The patterns are essentially the same as shown in Figure 8 but in the opposite direction across the interface.

In Eq. (23),  $\mathbf{j}$  describes the spatial location of a lattice site. After streaming on the coarse level and before advancing the fine level, the stream registers are initialized to the negative of distributions streaming from invalid coarse lattice sites,

$$\delta f_i^\ell(\mathbf{j}^\ell) = -f_i^\ell(\mathbf{j}^\ell, t^\ell + \Delta t^\ell). \quad (24)$$

For subcycle  $m$  on the coarse mesh, distributions streaming out from the fine grid are added to the register

$$\delta f_i^\ell(\mathbf{j}^\ell) = \delta f_i^\ell(\mathbf{j}^\ell) + \sum_{\mathbf{j}^{\ell+1} \in \mathcal{I}^{-1}(\mathbf{j}^\ell, m)} \hat{f}_i^{\ell+1}(\mathbf{j}^{\ell+1}, t^\ell + m\Delta t^{\ell+1}). \quad (25)$$

Here,  $\mathcal{I}^{-1}(\mathbf{j}^\ell, m)$ , denotes any interior lattice site on the fine grid that is adjacent to the interface, i.e.,

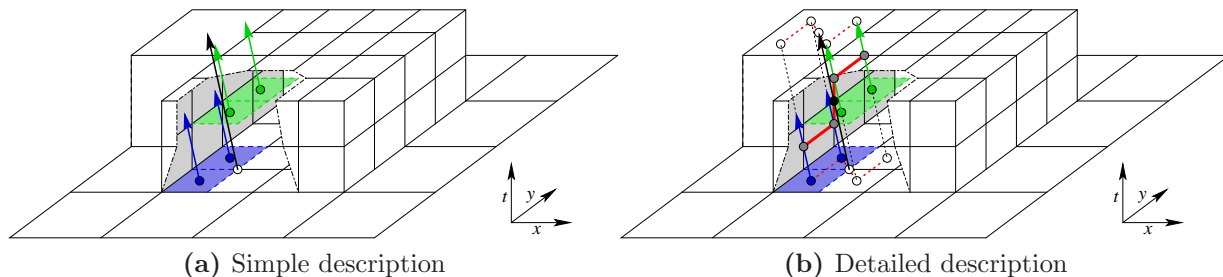
$$\mathbf{j}^{\ell+1} \in (\Omega^{\ell+1} - \mathcal{G}(\Omega^{\ell+1}, -1)) ,$$

and in which the trace of the stream vector in direction  $i$  at time  $t^\ell + \Delta t^\ell$  terminates in cell  $\mathcal{C}_{n_{\text{ref}}^\ell}^{-1}(\mathbf{j}^\ell)$ . Finally, once the fine grid reaches the same time as the coarse grid, the corrections are applied to the distributions on the coarse grid

$$f_i^\ell(\mathbf{j}^\ell, t^\ell + \Delta t^\ell) = f_i^\ell(\mathbf{j}^\ell, t^\ell + \Delta t^\ell) + \delta f_i^\ell(\mathbf{j}^\ell). \quad (26)$$

Definitions of  $\mathcal{I}^{-1}(\mathbf{j}^\ell, m)$  for various velocity directions and extents of fine boxes are shown in Figs. 15–16. The concepts are simply the reverse of those used for interpolation.

In Figure 16, stream correction across a face of the fine-grid box is shown for velocity direction  $(-x, +y)$ . For the same velocity direction, identifying the fine lattice sites that contribute to stream registers outside the  $(x_{\min}, y_{\max})$  corner of the fine-grid box — top left corner in Figure 16 — poses no extra challenge (tracing the fine stream vectors backwards would highlight 3 blue cells and 1 green cell in the corner). A much more interesting effect occurs at the  $(x_{\min}, y_{\min})$  convex corner — bottom left corner in Figure 16 — still for velocity



**Figure 16** Contributions to the stream register from the fine grid for streaming in edge-velocity direction  $(-x, +y)$ . In Figure 16a, lattice sites at the interior of the fine grid that contribute to the stream register associated with the black stream vector are shown in blue for the first subcycle and green in the second subcycle. The trace of fine stream vectors originating at these sites all terminate in a refinement of the cell in which the coarse stream vector terminates; this is how these sites were identified. The patterns are essentially the same as shown in Figure 10 but in the opposite direction across the interface.

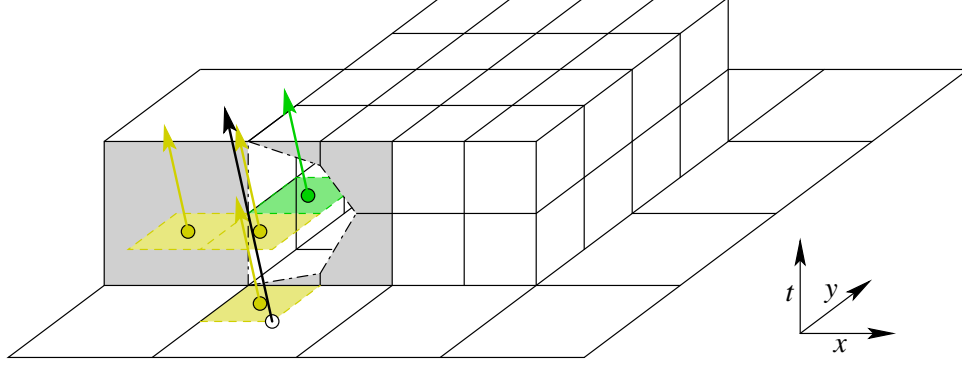
direction  $(-x, +y)$ . As shown in Figure 17, the streaming is transverse to the direction of the corner. If the black stream vector is divided into four, each originating at a refinement of the coarse cell (examine the traces shown in Figure 18), one fine stream vector would cross the fine grid before terminating in the coarse grid, whereas the other three would remain entirely in the space-time coarse region.

We already know how to deal with the stream vector that crosses the fine grid. Its contribution to the stream register comes from the fine lattice as normal. But where should the other contributions come from, those that do not cross the fine grid? The answer is found in the orphaned yellow ghost sites, filled by interpolation in Figure 12, but not having a fine grid to stream into. These ghost sites are included in Figure 17 showing that contributions to the stream register from the fine grid consist of both distributions streaming from the fine grid and orphaned interpolations. One can view the sum of the orphaned interpolations as the distribution of the coarse streaming vector minus the interpolated distributions that do enter the fine grid. Building the stream register in this fashion leads to full conservation at the convex corners.

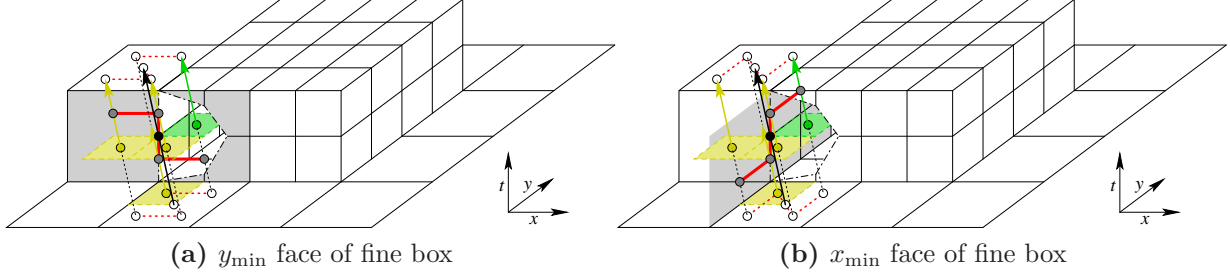
The same stream register is shown in Figure 18 with additional detail showing traces of the fine-grid stream vectors to times  $t^\ell$  and  $t^\ell + \Delta t^\ell$ . The interpretation of which interior fine cells and orphaned ghost sites contribute to the stream register is identical no matter if the interface is assumed to be at the  $y_{\min}$  face of fine box as shown in Figure 18a or the  $x_{\min}$  face of fine box as shown in Figure 18b. Both understandings of where the interface resides are valid at this convex corner of the fine grid.

A similar situation is observed at concave corners formed by the fine grid. In Figure 19, both interpolation (coarse-grid stream vector shown by the dashed black arrow) and stream corrections (coarse-grid stream vector shown by the solid black arrow) are illustrated for the  $(+x, +y)$  velocity direction. The interpolation is used to fill ghost sites shown by purple dots on the first subcycle and orange dots on the second subcycle. However, a problem arises in that there is no obvious conservative manner in which to interpolate to the ghost lattice site shown by the yellow dot. The fine grid contributions to the stream register associated





**Figure 17** Contributions to the stream register from the fine grid for streaming in edge-velocity direction  $(-x, +y)$ , but at a corner in the transfers direction,  $(x_{\min}, y_{\min})$  corner. Fine contributions to the stream register can only partially be obtained from the fine grid. The remaining contributions can be found in the orphaned yellow ghost sites, filled by interpolation in Figure 12.



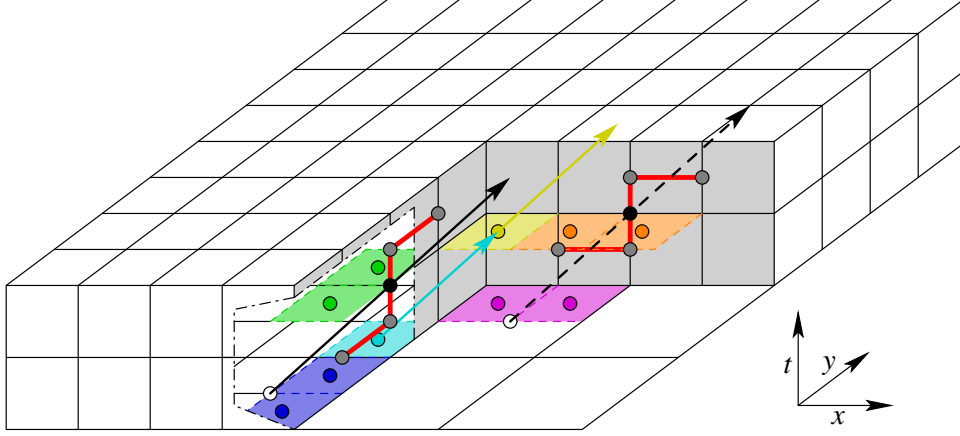
**Figure 18** Detailed description of fine-grid contributions to the stream register in a transverse velocity direction,  $(-x, +y)$ , at a convex corner of the fine grid. The origin and terminus of all streaming vectors are traced to times  $t^\ell$  and  $t^\ell + \Delta t^\ell$ . The same fine-grid contributions are identified if the interface is assumed at the  $y_{\min}$  face of fine box (Figure 18a) or the  $x_{\min}$  face of fine box (Figure 18b).

with the solid black arrow originate from the fine-grid lattice sites marked by blue dots on the first subcycle and green dots on the second subcycle. Another problem is observed in that the stream vector shown by the cyan arrow crosses into the coarse grid, but there is no stream register to associate it with. Fortunately, one problem is the solution to the other and vice-versa. In this situation, the distribution shown by the cyan arrow can be used to fill the yellow ghost-lattice site. This solution is conservative and ensures that a steady-state solution will not be perturbed.

### 3.6. Remarks on Implementation

The strategies for refinement of the lattice discussed herein compare most readily to the IVP approach advocated by Chen et al. [13]. Our proposed scheme is much more elaborate and a natural question to ask is “Is it really worth it?” Effects on accuracy will be discussed in the section on results. Here, we only comment on the difficulty of implementing the algorithm and the efficiency. The implementation is logistically complex without a doubt, perhaps favoring the approach in [13]. However, programming is greatly aided by a proper application





**Figure 19** Interpolation and stream corrections in a transverse  $(+x, +y)$  velocity direction at a concave corner of the fine grid. The interpolation is centered around a coarse-grid stream vector shown by the dashed black arrow. The stream correction is centered around a coarse-grid stream vector shown by the solid black arrow. There is no obvious conservative manner in which the fine-grid lattice site shown by the yellow dot can be filled by interpolation; there is also no stream register for the cyan stream vector to contribute towards. Instead, the distribution given by the cyan stream vector is used to set the ghost site at the yellow dot.

programming interface (API). As mentioned earlier, this application was built using the Chombo parallel AMR library. Chombo provides an extensive interface for manipulating “boxes” and storing data. All regions in which specific interpolation or stream correction patterns are required can be identified by refining, coarsening, shifting, intersecting, and taking unions of boxes (see Eq (23) for an example). Furthermore, operations on the boxes, as described above, are very inexpensive. By working with boxes, dealing with complex logistics becomes much more routine.

Despite the apparent complexity, the algorithm is still quite efficient. Identification of specific interpolation patterns need only occur during a regrid. Any matrices that must be inverted for the least-squares problems are cached at program start-up; computing an interpolant then only requires two matrix-vector multiplications. Filling a ghost site only involves a scalar product of two vectors. Furthermore, since these operations are only required at the interface, they are only performed on a codimension 1 subset of the full problem dimension. As with most time-variant AMR calculations, we find that most of the time is spent computing the solution on the finest lattice level.

#### 4. Results

We demonstrate our AMR algorithm and discuss the order of convergence and the errors associated with refinement through a series of benchmark problems. For the two flows with analytical solutions, transient Poiseuille, and the Taylor-Green vortex, the error was expressed as the solution error,  $u^n - u^a$ , where  $u^n$  is the computed solution and  $u^a$  is the exact solution. In the acoustic pulse example, the error was computed by taking the difference between the computed solution and a single grid calculation with a resolution twice that of the finest mesh of the computed solution. We compared the errors on refined grids to those

on single grids at several mesh resolutions. In addition, we also computed the errors for the algorithm proposed by Chen et al. [13] and Yu and Fan [29] to explore the differences between the initial-value and boundary-value problem approaches to populating the ghost cells at the coarse-fine interface. To mirror their approach, we reverted to a purely spatial piecewise linear coarse to fine interpolation using only the coarse cell distributions at  $t^\ell$  and incorporated two layers of ghost cells at interfaces. We did not include corrections that reduce the interpolation accuracy to preserve steady flows. As we only consider unsteady flows, our implementation of the IVP approach can be considered as more accurate than if we had rigorously followed the algorithm presented in [13]. Finally to highlight the adaptivity inherent in the Chombo library, we simulated the three-dimensional Taylor-Green transition and the Kármán vortex street shed by a circular cylinder. All simulations used the D3Q19 lattice.

#### 4.1. Transient Poiseuille flow

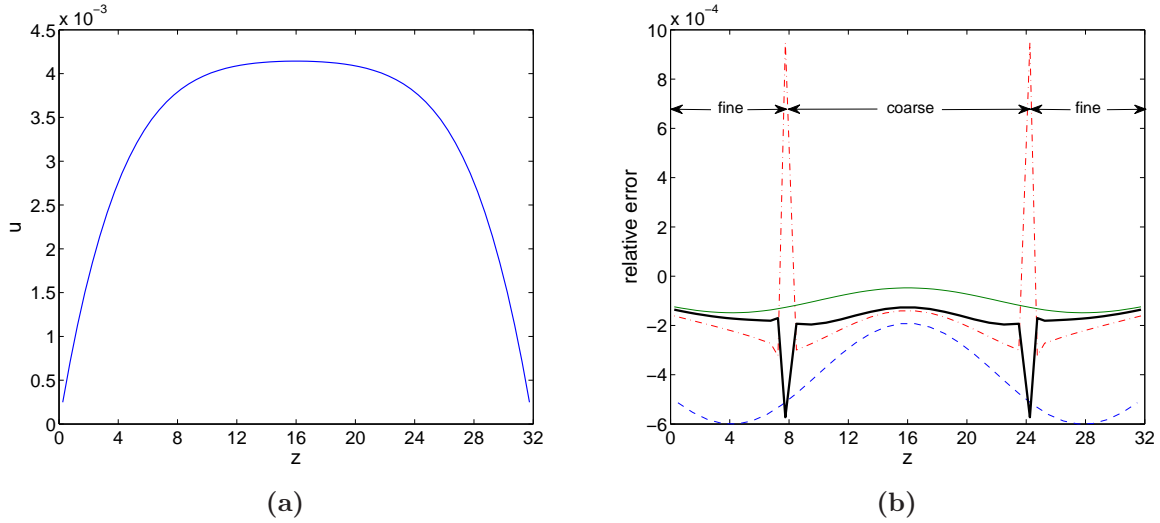
We first consider the accelerating flow resulting from the application of a constant pressure gradient to a fluid contained between two parallel plates. From rest, the flow evolves according to

$$u(z, t) = \frac{Gh}{2\mu}z \left(1 - \frac{z}{h}\right) - \frac{4Gh^2}{\mu\pi^3} \sum_{n=1,3,\dots}^{\infty} \frac{1}{n^3} \sin\left(\frac{n\pi z}{h}\right) \exp\left(-\frac{n^2\pi^2\nu t}{h^2}\right), \quad (27)$$

where  $G$  is the pressure gradient,  $h$  is the channel height, and  $z$  is the direction normal to the wall. The solution converges to a steady-state parabolic profile at long times. We chose  $h = 32$ ,  $\tau = 0.516$ , and a forcing of  $G = 1.042 \times 10^{-6}$ , which corresponds to a steady-state Reynolds number of 100. We ran the simulation to an intermediate state at  $t = 4,000$ , when the velocity profile is still fairly flat at the center of the channel. To minimize the error associated with the no-slip boundary condition, we employed the TRT collision operator with the second relaxation time tuned for planar boundaries located at cell edges. We ran the calculation on single grids and on two-level grids with a refinement ratio,  $n_{ref}^\ell = 2$ . The finer grids were located adjacent to the walls with a thickness  $h/4$  in the wall-normal direction and extended throughout the domain in the  $x$  and  $y$  directions.

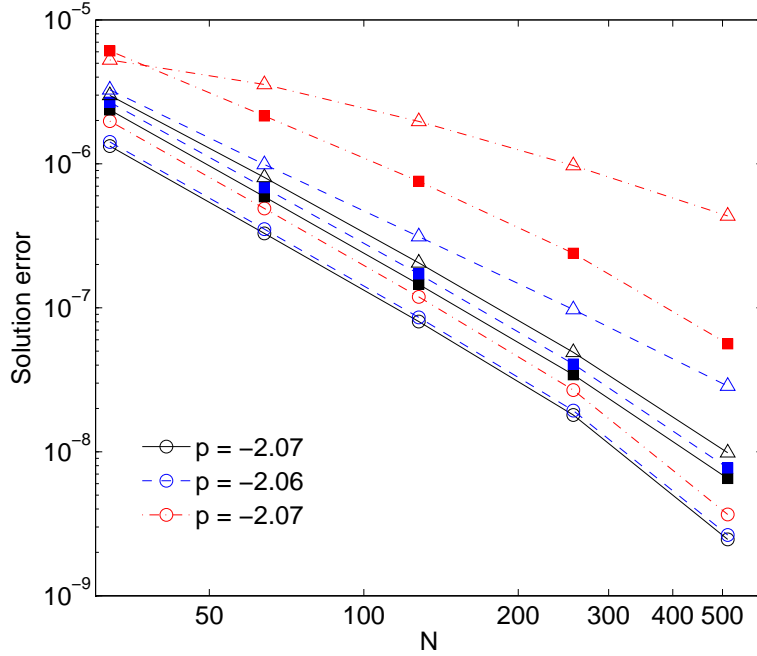
Figure 20 shows the analytical velocity profile,  $u^a$ , at time  $t = 4,000$ , and the relative error,  $e^{rel} = (u^n - u^a)/u^a$  for three different grids: a single coarse resolution with  $32 \times 4 \times 32$  cells in the  $x$ ,  $y$ , and  $z$  directions, a single fine resolution grid with  $64 \times 8 \times 64$  cells, and the two-level mesh where the coarser level is equivalent to the single coarse grid. The locations of the fine and coarse cells for the two-level mesh are indicated in the error plot; the finer cells cover the region with the largest shear rates. As expected, the single fine grid produced the smallest error across the channel. With both interpolation methods, the error on the finest level approached that of the single fine grid, whereas in the coarser region (the center of the channel) the error was somewhat smaller than on the single coarse grid. However there are differences in the errors of the two refinement approaches. Overall the BVP algorithm produced smaller errors than the IVP interpolation, especially near the coarse-fine interfaces.

Further comparisons between the two interpolation methods are provided in Figure 21, a plot of the solution error, expressed in  $L_1$ ,  $L_2$ , and  $L_\infty$ -norms, for single and two-level meshes as function of grid resolution,  $N$ , where the two-level grid resolution is defined by the finer



**Figure 20** a) Velocity profile at  $t = 4,000$ , b) Relative error profile of the computed velocity  $e^{rel} = (u^n - u^a) / u^a$ : single coarse grid (dashed blue line), single fine grid (solid green line), IVP interpolation (dash-dot red line), and BVP interpolation (thick solid black line).

level. The slopes,  $p$ , of the single level error curves are listed in the legend for reference. For a given resolution, the AMR errors are somewhat higher than those of the single grid; this is expected given that the grid is coarser at the center of the channel and the introduction of discontinuous changes in grid resolution. Generally, the  $L_1$  and  $L_2$ -norms smooth out the errors at the interface which are captured by the  $L_\infty$ -norm, and there are clear differences between the AMR and single grid slopes of this norm. These differences are quantified in Table 1, which lists in alternating columns the numerical values for the errors and the rates of convergence. As expected, the single grid is second order in all three norms, although it is interesting to note for this particular problem the convergence rates are close to three at the highest resolutions tested. The calculation of the slopes listed in the legend did not include the errors at the 512 resolution. Discontinuous changes in grid resolution, such as those introduced by AMR, can be expected to reduce the order of accuracy by one. In practice, however, the reduction in accuracy is often minimized, either by keeping the discontinuous grid changes well away from areas of interest or through advantageous countermanding features. An example of the latter, in the case of hyperbolic wave propagation, is that the region containing the error itself reduces by  $O(h)$  with refinement and cancels the reduction in accuracy as long as solution characteristics travel perpendicular to the refinement interface. Therefore, the BVP approach to AMR is still observed as second order in the  $L_1$  and  $L_2$ -norms, and even converges to second order in  $L_\infty$  as the mesh is refined. The IVP method is only second order in  $L_1$  and is reduced to first order in  $L_\infty$  due to the large errors at the interfaces. The IVP interpolation  $L_2$ -norm is roughly four times larger than the BVP interpolation, and the  $L_\infty$ -norm is eight times larger at the highest resolution. In the IVP approach, neglecting to incorporate temporal variations in the solution when populating the fine ghost cells at the interface has consequences for both the error magnitude and the overall order of method. Interpolating in time is important to maintain accuracy in unsteady flows.



**Figure 21** Solution errors as a function of grid resolution,  $N$ , for the transient Poiseuille flow at  $t = 4,000$  for single ( $\circ$ ) and two-level grids with IVP ( $\triangle$ ) and BVP ( $\blacksquare$ ) interpolation:  $L_1$ -norm (solid black),  $L_2$ -norm (dashed blue), and  $L_\infty$ -norm (dash-dot red). For the two-level meshes the resolution corresponds to the finer level. The legend contains the slopes,  $p$ , of the single-grid curves.

## 4.2. Taylor-Green vortex

### 4.2.1. Two-dimensional flow

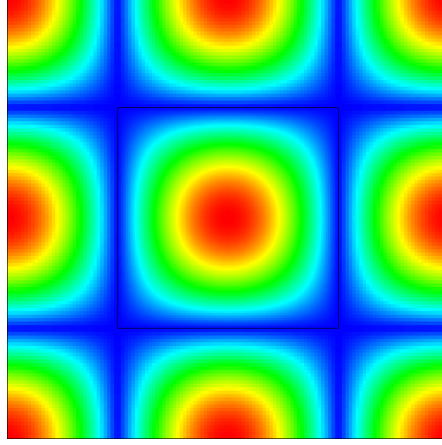
We first consider the two-dimensional Taylor-Green problem, a periodic array of stationary vortices whose strength decays in time for low Reynolds numbers. The solution for the velocity and density are given by

$$\begin{aligned} u &= -U_0 \exp(-2k^2 \nu t) \cos(kx) \sin(ky) \\ v &= U_0 \exp(-2k^2 \nu t) \sin(kx) \cos(ky) \\ \rho &= \rho_0 \left[ 1 - \frac{3}{4} U_0^2 \exp(-4k^2 \nu t) (\cos(2kx) + \cos(2ky)) \right], \end{aligned} \quad (28)$$

where  $U_0$  is the velocity amplitude,  $k = n\pi/L_0$ ,  $n$  is the number of vortices in each direction, and  $L_0$  is the length of the domain. For this benchmark we chose  $n = 2$ ,  $U_0 = 0.001$ ,  $L_0 = 128$ ,  $\rho_0 = 1.2$ , and  $\nu = 1/300$ , corresponding to a Reynolds number,  $Re = U_0 L_0 / \nu = 38$ . Similar to the previous problem, we ran on single and two-level grids at five different resolutions ( $64^2$ ,  $128^2$ ,  $256^2$ ,  $512^2$ , and  $1024^2$ ) for 600 coarse level time steps. There were four cells in the  $z$ -direction, normal to the flow, at each resolution. For the two-level calculations, the finer level, with a refinement ratio of 2, was centered around the middle vortex with lateral

**Table 1** Numerical values of the transient Poiseuille flow solution errors from Figure 21 and the convergence rates between consecutive grid resolutions.

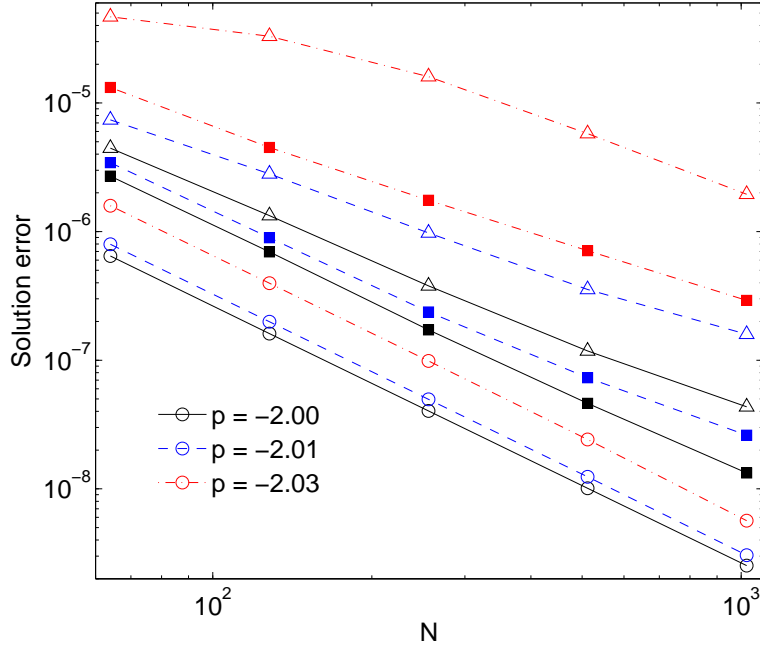
single	32	rate	64	rate	128	rate	256	rate	512
$L_1$	1.33e-06	2.02	3.29e-07	2.04	8.02e-08	2.16	1.80e-08	2.87	2.46e-09
$L_2$	1.42e-06	2.01	3.52e-07	2.04	8.57e-08	2.15	1.93e-08	2.87	2.64e-09
$L_\infty$	1.98e-06	2.02	4.90e-07	2.04	1.19e-07	2.15	2.68e-08	2.87	3.67e-09
IVP									
$L_1$	2.99e-06	1.90	8.05e-07	1.97	2.05e-07	2.06	4.91e-08	2.32	9.85e-09
$L_2$	3.27e-06	1.72	9.90e-07	1.67	3.12e-07	1.68	9.72e-08	1.76	2.86e-08
$L_\infty$	5.27e-06	0.57	3.56e-06	0.85	1.97e-06	1.02	9.73e-07	1.16	4.35e-07
BVP									
$L_1$	2.37e-06	2.01	5.90e-07	2.02	1.45e-07	2.08	3.43e-08	2.39	6.53e-09
$L_2$	2.68e-06	1.97	6.82e-07	1.99	1.72e-07	2.08	4.07e-08	2.40	7.72e-09
$L_\infty$	6.09e-06	1.50	2.15e-06	1.51	7.56e-07	1.66	2.39e-07	2.09	5.62e-08



**Figure 22** Taylor-Green vorticity field with  $n = 2$  at time  $t = 0$ . The level 1 (fine) mesh is centered around the middle vortex and outlined in black.

dimensions  $L_0/4$ , and a depth of  $4\Delta x$  (the fine mesh is contained within the black outline in Figure 22).

The Taylor-Green flow is not well-suited for AMR since the error is fairly uniform across the domain with the checkerboard pattern of vortex cores and stagnation points. However, the solution error highlights the interpolation errors at the interfaces and is therefore useful for comparing various approaches to AMR. The results are plotted and listed in Figure 23 and Table 2. Again the slopes,  $p$ , of the single level curves are shown in the legend and demonstrate second-order convergence of the solutions, whereas AMR adversely affected the order of the LBM. For all three error norms, the AMR calculations diverged from second order as the resolution increased (note the decrease in slope of the triangle and square labeled lines in Figure 23). Still there were differences between the interpolation approaches. The IVP interpolation diverged to first order in the  $L_2$ -norm and had higher errors in all norms



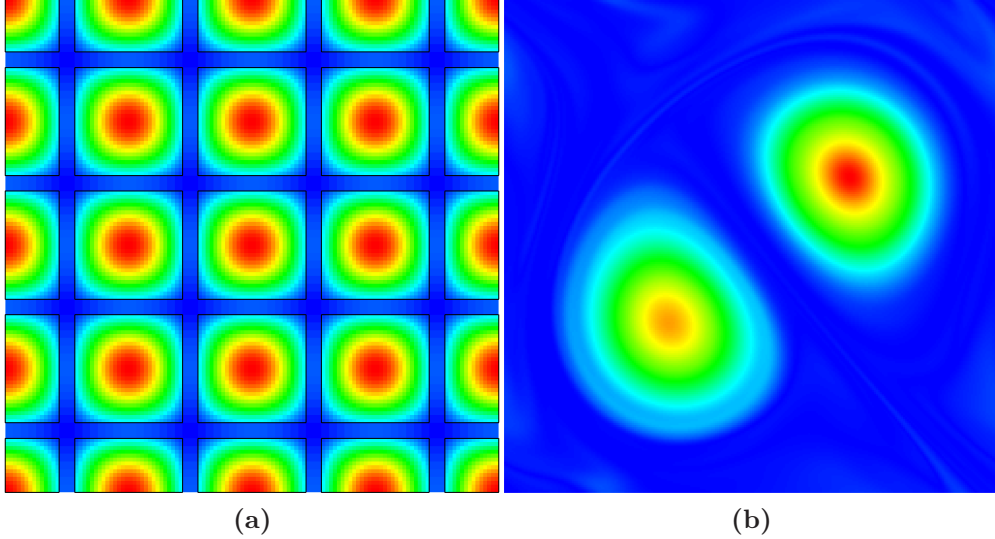
**Figure 23** Velocity solution errors as a function of grid resolution,  $N$ , for the Taylor-Green vortex at  $t = 600$  for single ( $\circ$ ) and two-level grids with IVP ( $\triangle$ ) and BVP ( $\blacksquare$ ) interpolations:  $L_1$ -norm (solid black),  $L_2$ -norm (dashed blue), and  $L_\infty$ -norm (dash-dot red). For the two-level meshes the resolution corresponds to the finer level. The legend contains the slopes,  $p$ , of the single-grid curves.

than the BVP approach. The IVP  $L_2$  errors were 2 – 6 times larger and the  $L_\infty$  errors were 4 – 9 times larger, depending on the resolution. Again these differences are most likely attributed to the lack of time interpolation in the IVP method.

When the Reynolds number is increased, the initial stable array of vortices can perturb itself and transition into a more unsteady but still two-dimensional flow. This behavior is demonstrated in Figure 24 which shows the initial array with  $n = 4$ , and at  $t = 20,000$ , after the vortices combined into a single counter-rotating pair that advected through the domain (the image in Figure 24b is from a  $256 \times 256 \times 4$  single grid calculation). In our simulations, we reduced the viscosity to obtain a  $Re = 64,000$  and used a non-adaptive two-level mesh locally refined around the initial positions of the vortex centers (Figure 24a). The coarse mesh resolution was  $64 \times 64 \times 4$ . With the high Reynolds number and transitioning flow, we could explore the stability of the various collision operators with fixed mesh refinement. In this context, stability means the positivity of the components of the distribution function,  $f_i$ . We ran simulations with the BGK, MRT, and the ELB collision operators: the BGK calculation became unstable after 5,000 time steps, the MRT calculation failed just shy of 10,000 time steps, and the ELB method remained stable throughout the entire run. Figure 25a shows the evolution of the domain H-function for the three collision operators. Though the interpolation and averaging operations do not necessarily guarantee a non-increasing numerical H-function on a multi-level mesh, the entropic method was still effective in main-

**Table 2** Numerical values of the Taylor-Green flow solution errors from Figure 23 and the convergence rates between consecutive grid resolutions.

single	64	rate	128	rate	256	rate	512	rate	1024
$L_1$	6.46e-07	2.00	1.61e-07	2.00	4.03e-08	2.00	1.01e-08	1.99	2.53e-09
$L_2$	7.96e-07	2.00	1.99e-07	2.00	4.96e-08	2.01	1.24e-08	2.02	3.05e-09
$L_\infty$	1.59e-06	2.00	3.97e-07	2.01	9.87e-08	2.03	2.42e-08	2.10	5.65e-09
IVP									
$L_1$	4.45e-06	1.74	1.33e-06	1.82	3.78e-07	1.68	1.18e-07	1.44	4.34e-08
$L_2$	7.38e-06	1.39	2.81e-06	1.53	9.76e-07	1.46	3.55e-07	1.16	1.59e-07
$L_\infty$	4.68e-05	0.51	3.30e-05	1.04	1.60e-05	1.47	5.78e-06	1.57	1.95e-06
BVP									
$L_1$	2.70e-06	1.96	6.96e-07	2.02	1.72e-07	1.89	4.62e-08	1.80	1.33e-08
$L_2$	3.43e-06	1.93	8.96e-07	1.92	2.37e-07	1.70	7.30e-08	1.48	2.61e-08
$L_\infty$	1.32e-05	1.55	4.51e-06	1.36	1.75e-06	1.30	7.10e-07	1.28	2.92e-07

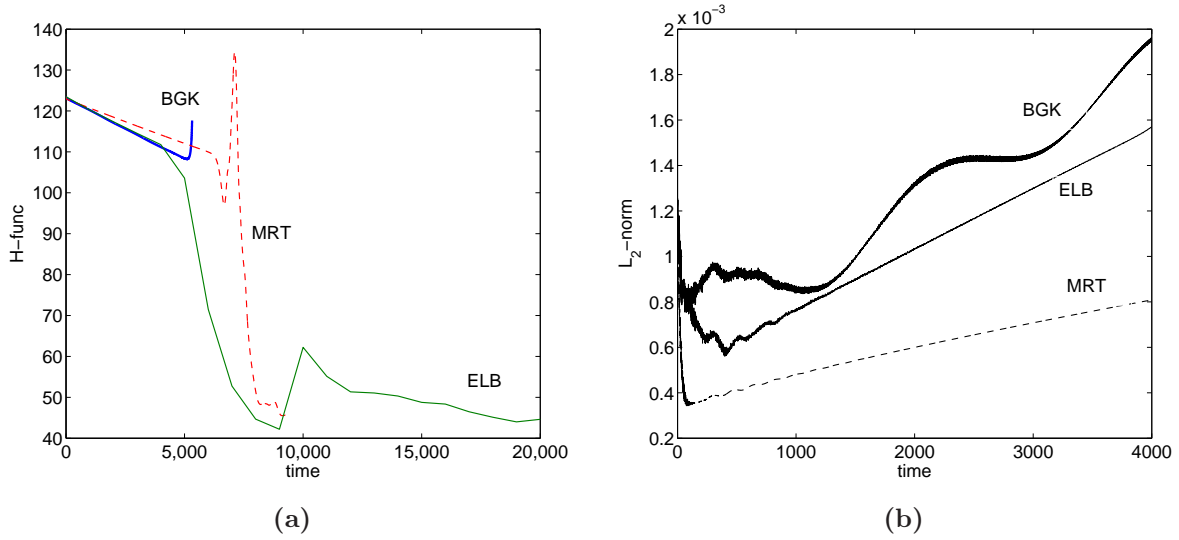


**Figure 24** Taylor-Green vorticity field with  $n = 2$  and  $Re = 64,000$ : a) at time  $t = 0$  on a two-level mesh with the finer level outlined with black rectangles, b) at time  $t = 20,000$ .

taining stability in our simulations. Of course, stability improves as the mesh is refined, which, from a practical standpoint, eliminates the necessity of the more expensive entropic method. Thus, one possible strategy is to use the ELB method on the coarser levels and revert to BGK collisions on the finer levels.

We also compared the solution errors of the three collision operators (Figure 25b) during the early stages of the simulations, before the collapse of the vortex array. All three simulations showed an increasing error with time since they eventually diverged from the analytical solution. The errors of the BGK and entropic relaxation methods were initially comparable, and the plot also demonstrates the effectiveness of the ELB method in removing the oscillations present in the BGK solution. Though it was ultimately unstable, the MRT





**Figure 25** Time evolution of the a) H-function, and b) solution error ( $L_2$ -norm) of the Taylor-Green flow at  $Re = 64,000$  for the BGK, MRT, and ELB collision operators.

operator produced the smallest error with the lowest divergence rate.

#### 4.2.2. Conservation

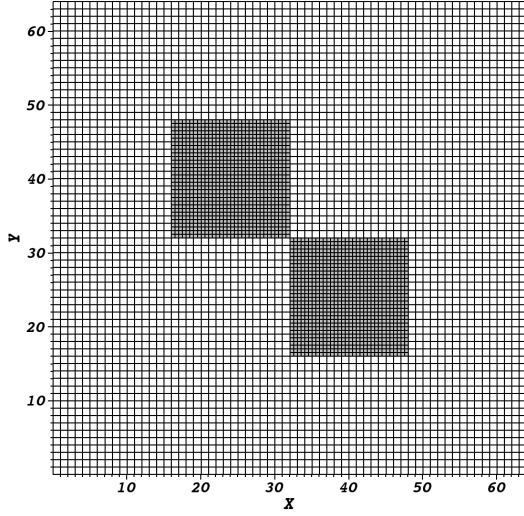
The periodic domain for the Taylor-Green vortex problem provides an opportunity to test conservation of the particle distributions in the AMR framework. Tests are performed in a cubic grid with two refined patches, also cubic in shape, as shown in Figure 26. The patches either touch at a corner (Figure 26a) to test a complex convex-corner scenario or partway along a face (Figure 26b) to test the concave-corner interpolation algorithms; in the orthogonal direction ( $z$ -direction in Figure 26), the fine patches are centered in the coarse grid. The coarse grid has  $64^3$  cells and the refinement ratio between the two grid levels is 2. An integration of the particle distributions in the valid cells on all levels is performed at the start of the run and after 1000 time steps. The double precision results are shown in Table 3. Although these results indicate that mass is conserved (with accommodation for round-off

**Table 3** Initial and final total particle density for Taylor-Green problem

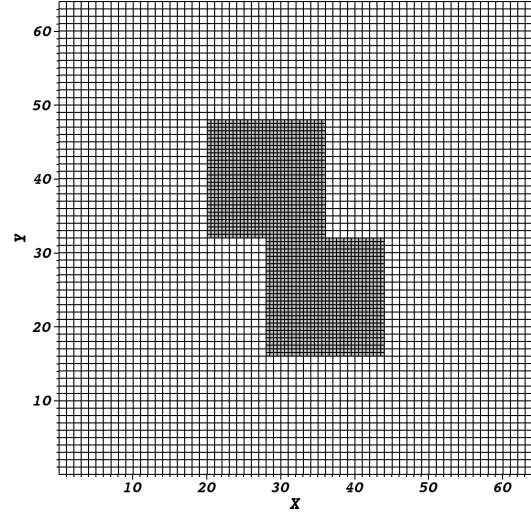
Test Case	Initial	Final	Difference
Figure 26a	2.621440000000002e+05	2.621439999999852e+05	0.000000000000150e+05
Figure 26b	2.621440000033370e+05	2.621440000033218e+05	0.000000000000152e+05

error), they say nothing about conservation of momentum. However, from the convergence rates reported in Table 2, we speculate that errors in momentum conservation converge at second order.





(a) Two-level grid for testing convex corners



(b) Two-level grid for testing concave corners

**Figure 26** Two-level grids for testing conservation of particle distributions across interfaces between coarse and fine grids. Each grid forms a cube in three dimensions; the fine patches are also cubic and are centered in the coarse grid in the  $z$ -direction.

#### 4.2.3. Three-dimensional flow

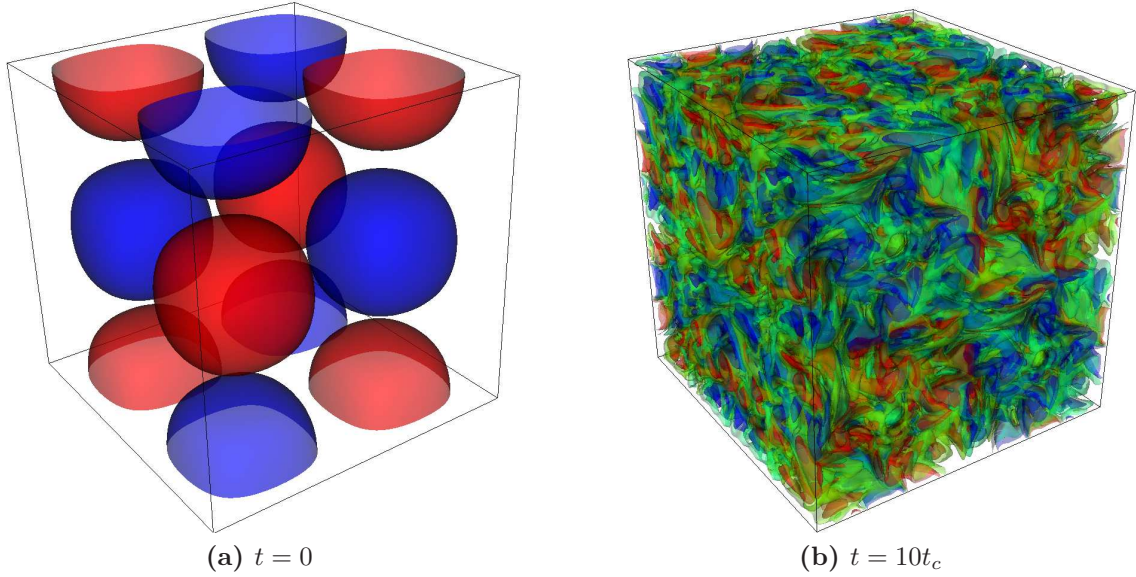
In the three-dimensional Taylor-Green problem, the initial array breaks down and transitions to a turbulent flow, producing a range of smaller scales through vortex stretching. Without an external source to maintain it, the flow eventually decays. With the initial conditions

$$\begin{aligned}
 u &= U_0 \sin(kx) \cos(ky) \cos(kz) \\
 v &= -U_0 \cos(kx) \sin(ky) \cos(kz) \\
 w &= 0 \\
 \rho &= \rho_0 \left[ 1 + \frac{3}{16} U_0^2 (\cos(2kx) + \cos(2ky)) (\cos(2kz) + 2) \right],
 \end{aligned} \tag{29}$$

where again,  $k = 2\pi/L_0$ , we ran simulations with a  $256^3$  single grid, and a fully adaptive grid with a  $128^3$  coarse level and one level of refinement. The refinement criterion was based on the local vorticity magnitude exceeding a threshold derived from the maximum vorticity. The flow evolved for ten characteristic times,  $t_c = \frac{L_0}{2\pi U_0}$ , and Figure 27 shows the initial and final vorticity fields. The kinetic energy dissipation  $\epsilon = -\frac{dE}{dt}$ , where

$$E = \frac{1}{\rho_0 \Omega^0} \int_{\Omega} \rho \frac{\mathbf{u} \cdot \mathbf{u}}{2} d\Omega, \tag{30}$$

of this flow has a well-defined peak and in Figure 28 we compare the results of our calculations with those from a pseudo-spectral method simulation at a much higher resolution ( $768^3$ ) [30]. Both the single grid and AMR produced dissipation evolution curves in excellent agreement



**Figure 27** Initial and final isosurfaces of the  $z$ -component of vorticity in the three-dimensional Taylor-Green vortex benchmark.

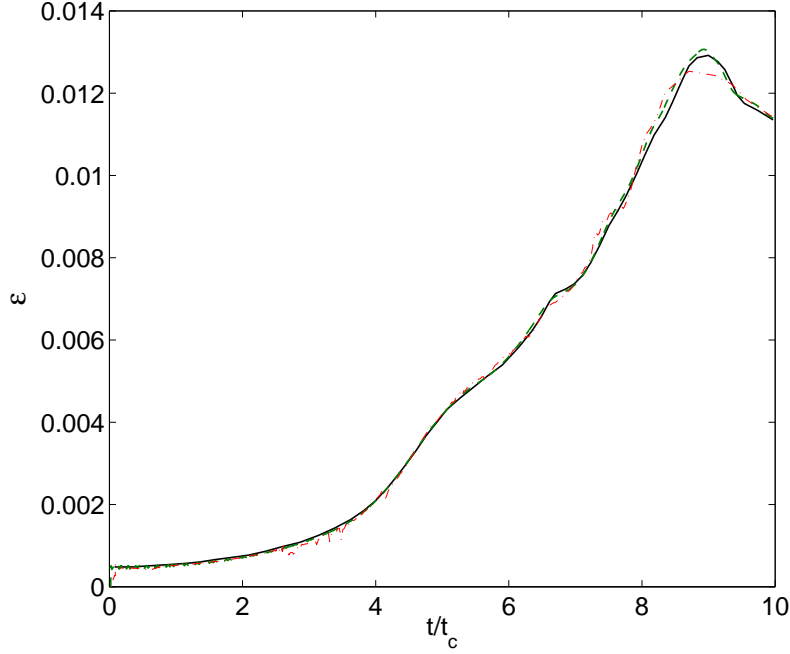
with the spectral method. There was some high frequency noise in the AMR data from the periodic regridding of the level 1 mesh which is likely responsible for the slight reduction in the peak dissipation but the other details of the curve are nicely reproduced.

#### 4.3. Acoustic Pulse

This benchmark, an example from gas dynamics, is the propagation of a Gaussian acoustic pulse in a periodic cubic domain with length  $L = 16$ . The initial velocity is zero and the density is given by

$$\rho(r) = \begin{cases} \rho_0 + (\delta\rho_0) \exp(-16r^2) \cos^6(\pi r), & r \leq \frac{L}{2}; \\ \rho_0 & \text{otherwise;} \end{cases} \quad (31)$$

where  $r$  is the distance from the center of the domain,  $\rho_0 = 1.4$ , and  $\delta\rho_0 = 0.14$ . We ran this problem on single and two-level grids with a refinement factor of two for four time steps. The finer level covered a cube with sides  $L/2$  at the center of the domain. Figure 29 shows the initial and final density fields. Since there is no analytical solution available for this problem, the error was defined as the difference in density between a given simulation and a single grid calculation at twice the resolution, where again the resolution of a two-level mesh is determined by the finer level. As before, these errors are displayed graphically and tabulated in Figure 30 and Table 4. In this benchmark, the IVP interpolation produced slightly smaller errors in the  $L_1$  and  $L_2$ -norms than the BVP interpolation. Both methods were also second order in these two norms but as the mesh was refined, the IVP  $L_\infty$  convergence rate increased somewhat then dropped to close to first order in the highest resolution (256) simulation. This behavior is easily seen from Figure 30 where the red curve with triangles (IVP) crosses over

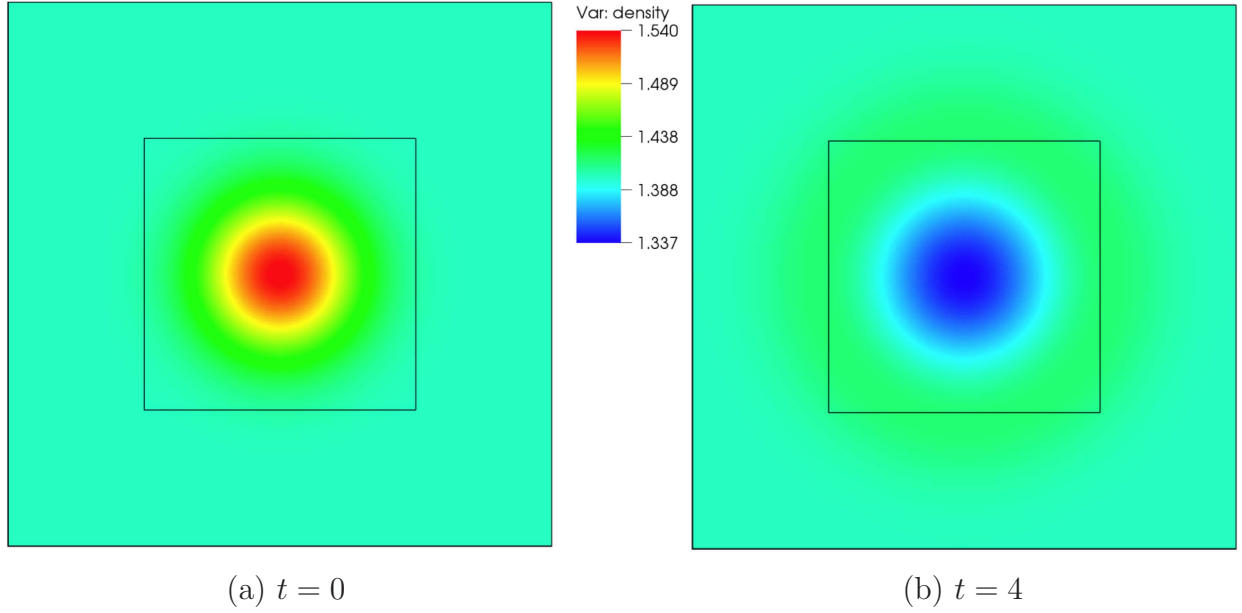


**Figure 28** Evolution of the 3D Taylor-Green normalized kinetic energy dissipation: pseudo-spectral method (solid black) [30], LB  $256^3$  single grid (dashed green), and LB AMR (dash-dot red)

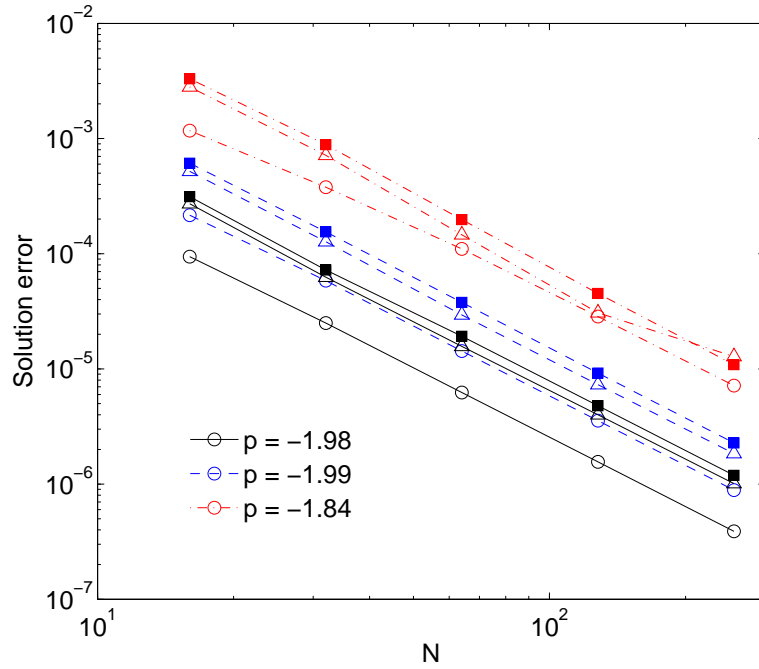
**Table 4** Numerical values of the acoustic pulse solution (density) errors from Figure 30 and the convergence rates between consecutive grid resolutions.

single	16	rate	32	rate	64	rate	128	rate	256
$L_1$	9.43e-05	1.91	2.50e-05	2.01	6.24e-06	2.00	1.56e-06	2.00	3.89e-07
$L_2$	2.16e-04	1.89	5.84e-05	2.03	1.43e-05	2.01	3.55e-06	2.00	8.87e-07
$L_\infty$	1.17e-03	1.63	3.78e-04	1.78	1.10e-04	1.96	2.85e-05	1.99	7.17e-06
IVP									
$L_1$	2.72e-04	2.12	6.26e-05	2.00	1.57e-05	1.97	4.00e-06	1.99	1.01e-06
$L_2$	5.20e-04	2.04	1.27e-04	2.11	2.94e-05	2.01	7.29e-06	1.99	1.84e-06
$L_\infty$	2.81e-03	1.97	7.16e-04	2.29	1.46e-04	2.25	3.07e-05	1.26	1.28e-05
BVP									
$L_1$	3.13e-04	2.12	7.22e-05	1.91	1.92e-05	2.00	4.79e-06	2.01	1.19e-06
$L_2$	6.11e-04	1.98	1.55e-04	2.04	3.78e-05	2.04	9.21e-06	2.01	2.28e-06
$L_\infty$	3.30e-03	1.90	8.85e-04	2.16	1.98e-04	2.13	4.52e-05	2.05	1.09e-05

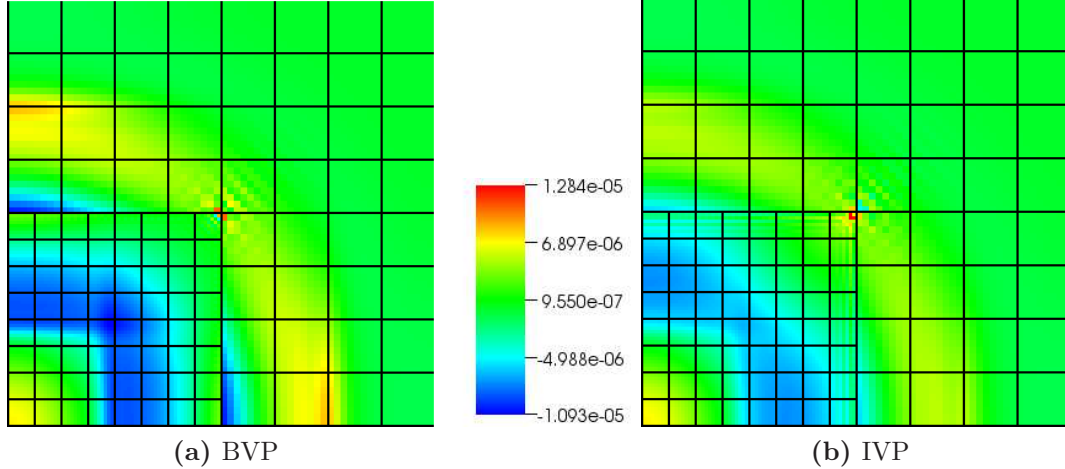
the adjacent line with solid squares (BVP). A more detailed look at the error fields (Figure 31) revealed the maximum error in the IVP approach occurred in the fine mesh just inside the corner of the interface. In contrast, the error of BVP interpolation was on average higher but the more robust handling of edge distributions at the interface corners maintained the second order accuracy of the  $L_\infty$ -norm.



**Figure 29** Two-dimensional slices of the density in the acoustic pulse benchmark. The boundary of the coarse-fine interface is outlined by the square in the center.



**Figure 30** Acoustic pulse density differences between consecutive grid resolutions at  $t = 4$  for single ( $\circ$ ) and two-level grids with IVP ( $\triangle$ ) and BVP ( $\blacksquare$ ):  $L_1$ -norm (solid black),  $L_2$ -norm (dashed blue), and  $L_\infty$ -norm (dash-dot red). For the two-level meshes the resolution,  $N$ , corresponds to the finer level. The legend contains the slopes,  $p$ , of the single-grid curves.



**Figure 31** Differences in density between the  $64^3$  refined two-level grid and the  $256^3$  single level grid. Boxes on the coarse and fine grids are outlined.

#### 4.4. Kármán Vortex Street

Another demonstration of adaptive mesh refinement is made for the classic case of vortex shedding behind a circular cylinder at Mach 0.05 and a Reynolds number of 100. The alternating shedding of the vortices produces a Kármán vortex street as shown in Figure 32. The complete problem domain is  $80 \times 32$  cylinder diameters, with the top, bottom, and inflow boundaries 16D from the center of the cylinder and the outflow boundary 64D from the cylinder center. The solution is three-dimensional but with only two coarse grid cells in the axial direction of the cylinder. The coarse grid otherwise features  $200 \times 80$  cells. The solution is obtained on a three-level grid with a refinement ratio of four between levels and the finer levels adaptively refined to follow regions of high vorticity magnitude.

On the finest grid, 40 cells span the diameter of the cylinder. Although the finest resolution still poorly resolves the growth of the boundary layer, it is sufficient to recover the robust quantities of Strouhal number,

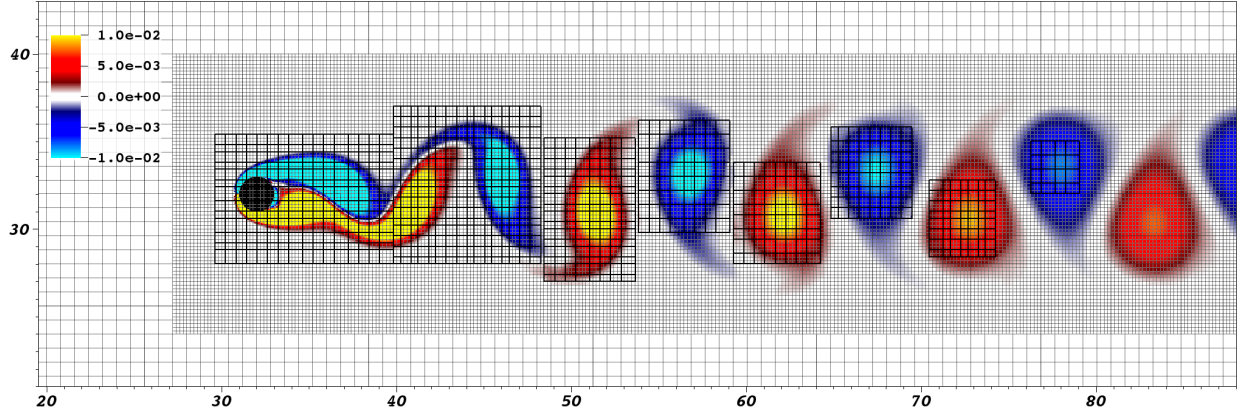
$$St \equiv \frac{fD}{u_\infty}, \quad (32)$$

and coefficient of base pressure (which can instead be measured by density since the fluid is isothermal)

$$C_{pb} \equiv \frac{(\rho_b - \rho_\infty)c_s^2}{\frac{1}{2}\rho_\infty u_\infty^2}. \quad (33)$$

In the above,  $f$  is the frequency for the vortex shedding and  $D$  is the diameter of the cylinder. Table 5 provides a comparison between our measured values and values resulting from physical experiments provided by Williamson and Roshko [31]. The plus/minus error estimate is taken from the spread of experimental data points.

The solution is not perfect and errors introduced by changes in the grid resolution are quite visible. Contours of vorticity are sensitive to changes in the grid and this is compounded by the eventual production of fine grids that are separated by one coarse cell. As the magnitude of vorticity diminishes downstream, the individual vortices are resolved by separate



**Figure 32** Illustration of vortex shedding behind a cylinder with the grid adaptively refined to follow regions of high vorticity magnitude. The flow is at Mach 0.05 and at a Reynolds number of 100. There are three grid levels with a refinement ratio of four between the levels. On the coarsest grid, 2.5 cells span the diameter of the cylinder; on the finest grid, 40 cells span the diameter. On the finest grid level, only the boxes are outlined, each containing at most  $16^3$  cells. These boxes are distributed among the processors to achieve an efficient parallel algorithm.

**Table 5** Comparison of numerical and experimental features for vortex shedding behind a cylinder at  $Re=100$ .

Number	Numerical	Experiment
$St$	0.166	$0.165 \pm 0.005$
$C_{pb}$	-0.69	$-0.72 \pm 0.02$

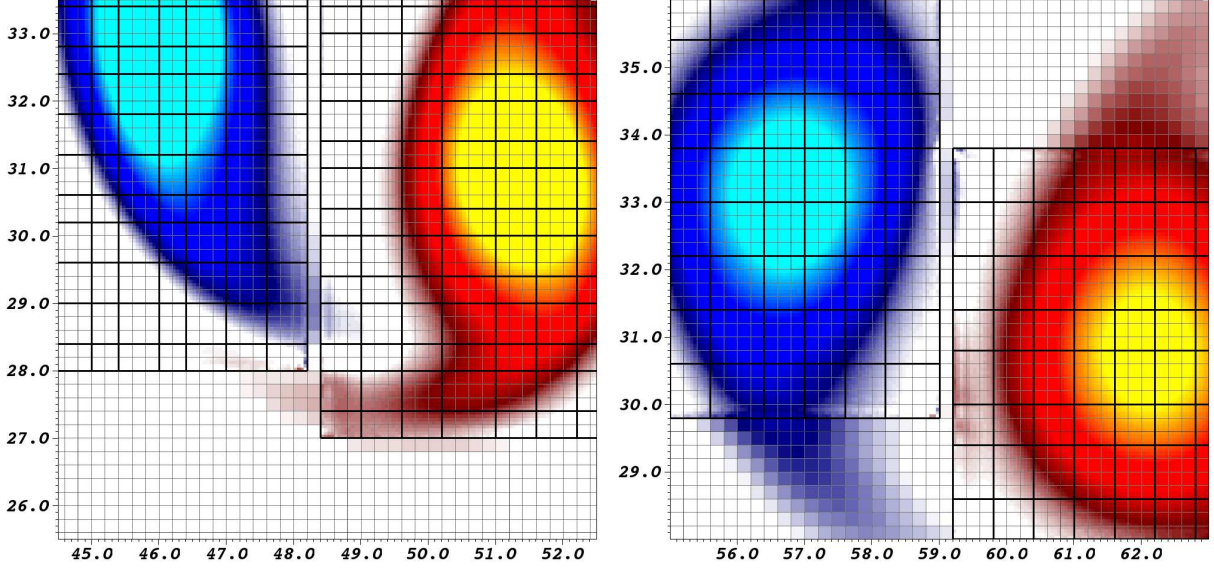
fine-grid patches. Initially, the separation is by one coarse cell, an inevitable situation in which the layer of coarse cells does little more than inject error into the solution. The errors are quite noticeable in Figure 33 which provides closer views of Figure 32. Errors at convex corners are quite interesting, showing both positive and negative deviations that result from balancing gradients in the  $+y$  and  $-x$  directions in order to maintain conservation.

## 5. Conclusions

We introduced a novel volume-based approach for adaptive mesh refinement for the lattice-Boltzmann method. Beginning with a discussion of the advantages of populating ghost cells as a boundary-value problem instead of as an initial-value problem, we proposed a least squares space-time interpolation algorithm with an imposed mass conservation constraint. Distributions corresponding to edge lattice velocities required special handling to maintain conservation during coarse to fine interpolation, and the coarse lattice velocity distributions that partially stream over fine cells were also adjusted to ensure mass conservation.

The strategies described herein for the D3Q19 lattice luckily ignore the problem of interpolating in corner velocity directions,  $\|\mathbf{e}_i\|_1 = 3$ . These velocity directions only exist on





**Figure 33** Closer views of Figure 32 in locations where fine grids are separated by one coarse cells. At these locations, errors introduced by the changes in grid resolution are quite visible.

lattices such as D3Q15 and D3Q27; in space-time, they are four-dimensional and difficult to visualize. However, determination of which ghost lattice sites are filled to conserve a given coarse stream vector, and which stream registers are corrected by fine lattice sites, can be understood from the traces of the fine stream vectors to coarse cells at times  $t^\ell$  and  $t^\ell + \Delta t^\ell$ . These traces were shown in the detailed drawings of the interpolation and stream correction patterns. We expect that similar relations can also be made for corner velocity directions and would assist in the development of interpolation and stream correction patterns for lattices with these velocity directions.

We compared our approach with initial-value lattice-Boltzmann mesh refinement featuring linear spatial interpolation through a series of benchmarks. A detailed analysis of the solution error and rate of convergence revealed the advantages of our method. The errors at the coarse-fine grid interface, quantified by the  $L_\infty$ -norm, produced by the boundary-value interpolation were smaller than and had a higher convergence rate than the errors of the initial-value scheme. With the exception of the two-dimensional Taylor-Green problem, the BVP interpolation maintained the second order convergence of the lattice-Boltzmann method. It is likely that additional gains in accuracy are possible but only at the cost of sacrificing conservation. However, it was noted that the increase in accuracy of the BVP approach comes at the expense of significantly increased algorithmic complexity compared to the IVP approach.

By implementing the new algorithms within the Chombo AMR infrastructure, we could leverage the mesh adaptivity and embedded boundary capabilities, as well as the high level of parallelism already inherent to the library. Both these features were demonstrated by the Kármán vortex street simulation, and the adaptivity was also included in the three-dimensional Taylor-Green simulations.

## Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

## References

- [1] S. Succi, *The Lattice Boltzmann Equation*, Oxford Science Publications, 2001.
- [2] H. D. Yu, S. S. Girimaji, L. S. Luo, Lattice Boltzmann simulations of decaying homogeneous isotropic turbulence, *Phys. Rev. E* 71 (2005) 016708.
- [3] D. S. Clague, B. D. Kandhai, R. Zhang, P. M. A. Sloot, Hydraulic permeability of (un)bounded fibrous media using the lattice Boltzmann method, *Phys. Rev. E* 61 (2000) 616–625.
- [4] J. Zhang, P. C. Johnson, A. S. Popel, Red blood cell aggregation and dissociation in shear flows simulated by lattice Boltzmann method, *J. Biomech.* 41 (2008) 47–55.
- [5] R. W. Mei, W. Shyy, On the finite difference-based lattice Boltzmann method in curvilinear coordinates, *J. Comput. Phys.* 143 (1998) 426–448.
- [6] G. W. Peng, H. W. Xi, C. Duncan, S. H. Chou, Finite volume scheme for the lattice Boltzmann method on unstructured meshes, *Phys. Rev. E* 59 (1999) 4675–4682.
- [7] O. Filippova, D. Hanel, Grid refinement for lattice-BGK models, *J. Comput. Phys.* 147 (1998) 219–228.
- [8] A. Dupuis, B. Chopard, Theory and applications of an alternative lattice Boltzmann grid refinement algorithm, *Phys. Rev. E* 67 (2003) 066707.
- [9] D. Kandhai, W. Soll, S. Chen, A. Hoekstra, P. Sloot, Finite-Difference Lattice-BGK methods on nested grids, *Comp. Phys. Comm.* 129 (2000) 100–109.
- [10] D. Yu, R. Mei, W. Shyy, A multi-block lattice boltzmann method for viscous fluid flows, *Int. J. Numer. Meth. Fluids* 39 (2002) 99–120.
- [11] Y. Peng, C. Shu, Y. T. Chew, X. D. Niu, X. Y. Lu, Application of multi-block approach in the immersed boundary-lattice Boltzmann method for viscous fluid flows, *J. Comput. Phys.* 218 (2006) 460–478.
- [12] D. Lagrava, O. Malaspinas, J. Latt, B. Chopard, Advances in multi-domain lattice Boltzmann grid refinement, *J. Comput. Phys.* 231 (2012) 4808–4822.
- [13] H. Chen, O. Filippova, J. Hock, K. Molvig, R. Shock, C. Teixeira, R. Zhang, Grid refinement in lattice Boltzmann methods based on volumetric formulation, *Physica A* 362 (2006) 158–167.



- [14] M. Rohde, D. Kandhai, J. J. Derksen, H. E. A. van den Akker, A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes, *Int. J. Numer. Meth. Fluids* 51 (2006) 439–468.
- [15] P. Colella, D. T. Graves, N. Keen, T. J. Ligocki, D. F. Martin, P. McCorquodale, D. Modiano, P. Schwartz, T. Sternberg, B. V. Straalen, Chombo Software Package for AMR Applications - Design Document, Lawrence Berkeley National Laboratory, 2009. <https://seesar.lbl.gov/anag/chombo/ChomboDesign-3.0.pdf>.
- [16] M. J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989) 64–84.
- [17] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases .1. Small amplitude processes in charged and neutral one-component systems, *Phys. Rev.* 94 (1954) 511–525.
- [18] C. S. Sunder, G. Baskar, V. Babu, D. Strenski, A detailed performance analysis of the interpolation supplemented lattice Boltzmann method on the Cray T3E and Cray X1, *Int. J. High Perform. Comp.* 20 (2006) 557–570.
- [19] S. Chen, G. D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* 30 (1998) 329–364.
- [20] P. Lallemand, L. S. Luo, Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability, *Phys. Rev. E* 61 (2000) 6546–6562.
- [21] D. d’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, L. S. Luo, Multiple-relaxation-time lattice Boltzmann models in three dimensions, *Philos. Transact. A Math Phys. Eng. Sci.* 360 (2002) 437–451.
- [22] B. Chun, A. J. C. Ladd, Interpolated boundary condition for lattice Boltzmann simulations of flows in narrow gaps, *Phys. Rev. E* 75 (2007) 066705.
- [23] R. Adhikari, K. Stratford, M. E. Cates, A. J. Wagner, Fluctuating lattice Boltzmann, *Europhys. Lett.* 71 (2005) 473–479.
- [24] I. Ginzburg, D. d’Humières, Multireflection boundary conditions for lattice Boltzmann models, *Phys. Rev. E* 68 (2003) 066614.
- [25] B. Duenweg, A. J. C. Ladd, Lattice Boltzmann Simulations of Soft Matter Systems, in: Holm, C and Kremer, K (Ed.), *Advanced Computer Simulation Approaches for Soft Matter Sciences III*, volume 221 of *Advances in Polymer Science*, 2009, pp. 89–166.
- [26] S. Ansumali, I. V. Karlin, H. C. Ottinger, Minimal entropic kinetic models for hydrodynamics, *Europhys. Lett.* 63 (2003) 798–804.
- [27] S. S. Chikatamarla, S. Ansumali, I. V. Karlin, Entropic lattice Boltzmann models for hydrodynamics in three dimensions, *Phys. Rev. Lett.* 97 (2006) 010201.

- [28] M. J. Berger, On conservation at grid interfaces, SIAM J. Numer. Anal. 24 (1987) 967–984.
- [29] Z. Yu, L.-S. Fan, An interaction potential based lattice boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation, J. Comput. Phys. 228 (2009) 6456–6478.
- [30] W. M. van Rees, A. Leonard, D. I. Pullin, P. Koumoutsakos, A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers, J. Comput. Phys. 230 (2011) 2794–2805.
- [31] C. H. K. Williamson, A. Roshko, Measurements of base pressure in the wake of a cylinder at low reynolds numbers, Z. Flugwiss. Weltraumforsch. 14 (1990) 38–46.